NASA Contractor Report 172567

NASA-CR-172567
19850013692

EARTH RADIATION BUDGET EXPERIMENT SOFTWARE DEVELOPMENT

William L. Edmonds

Systems and Applied Sciences Corporation
17 Research Drive
Hampton, VA 23666

Contract NAS1-16571

April 1985

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

TABLE OF CONTENTS

ABSTRACT

Computer programming and analysis efforts during this three phase contract were carried out in support of the Earth Radiation Budget Experiment (ERBE) at NASA/Langley. The Earth Radiation Budget Experiment is described as well as data acquisition, analysis and modelling support for the testing of ERBE instruments. Also included are descriptions of the programs developed to analyze, format and display data collected during testing of the various ERBE instruments. Listings of the major programs developed under this contract are located in an appendix.

SECTION 1 - INTRODUCTION

Computer programming and analysis efforts during this three phase contract were carried out in support of the Earth Radiation Budget Experiment (ERBE) at NASA/Langley. The Earth Radiation Budget Experiment will be described in Section 2 of this report. Data acquisition, analysis and modelling support for the testing of ERBE instruments will be described in Section 3 of this report. In Section 4 will be found descriptions of the programs developed to analyze, format and display data collected during testing of the various ERBE instruments. Listings of major programs developed under this contract will be found in an appendix to this report.

SECTION 2 - ERBE

The Earth Radiation Budget Experiment is designed to measure the energy reflected by and emitted by the whole Earth. Three separate satellites will be placed in orbit carrying sets of instruments to measure the Earth's radiative "budget". The data collected during the lifetime of these satellites will facilitate a better understanding of our weather and climate on both a global and local scale.

At the time of this report, two of the three satellites carrying ERBE instruments were already in orbit. The first of these was ERBS (Earth Radiation Budget Satellite) launched from the Space Shuttle in October 1984. The second set of instruments was launched from Vandenburg Air Force Base in California aboard a TIROS near-polar orbital satellite in December, 1984. The third and final set of instruments will be on another TIROS/Vandenburg launch in late 1985. It is hoped that the three separate orbits achieved will give maximum coverage of the Earth's radiation budget in all areas of the globe.

The ERBE instrument sets each include a Scanner instrument and a Non-Scanner instrument. The Scanner instrument has three detectors which are used to measure radiation in the long wave, short wave and total regions of the spectrum. The Non-Scanner instrument contains radiometers for medium field of view and wide field of view, each making measurements in the long wave and total regions of the spectrum. The Non-Scanner also has an instrument called a Solar Monitor for measuring the Sun's

intensity.

Calibration of highly sophisticated instruments like the ERBE Scanner and Non-Scanner is an important and demanding part of a successful scientific investigation. The ERBE instruments were designed to be self calibrating in orbit in order to take into account any gradual changes with time of the various instrument response functions. The Non-Scanner can be rotated to view space as a nearly zero source of radiation, and then rotated to face the Sun as a known very high level source of radiation. The Scanner views space briefly during every four second scan to provide a zero reference. The Scanner also has the capability to view the sun through its Mirror Attenuator Mosaic (MAM). The MAM reduces the intensity of the incoming radiation to a safe level which will not damage the Scanner detectors. Each instrument also has a blackbody radiation source for internal calibration. Taken together, these calibration sources provide a comprehensive means to assure the quality of all ERBE data telemetered to Earth.

SECTION 3 - DATA ACQUISITION, ANALYSIS AND MODELLING

The design, fabrication and testing of the ERBE engineering-preflight- and flight-models required considerable use of computers. Radiometric and thermal modelling, data acquisition, recording and analysis are major areas of computer usage in the ERBE project.

Radiometric and Thermal Modelling

Radiometric and thermal modelling of the ERBE radiometers involved a number of programming efforts on different computers located within NASA/Langley and at the various instrument and satellite contractors' facilities. Support under this contract was given to several of these modelling efforts. Radiometric and thermal modelling software developed by Dr. Robert Mahan of VPI&SU was converted to a form usable on the NASA/Langley CDC computer system. Thermal modeling efforts at TRW were augmented by the utilization of the Vector-Instruction-Set of the HP-1000 computer system. Computer calculations which had taken as long as sixteen hours to perform were reduced to less than one minute without a loss of accuracy.

Data Acquisition and Recording

Data acquisition and recording software and facilities were provided by the instrument and satellite manufacturing contractors. During instrument testing, satellite integration and pre-launch satellite testing, support was given in monitoring

the acquisition and recording of data. Set-up and check-out of the HP-1000 computer at the Western Test Range (Vandenburg Air Force Base, California) was done in preparation for pre-launch testing of the TIROS/ERBE satellite. Support was given to the thermal-vac testing of ERBS (Earth Radiation Budget Satellite) at Ball Aerospace in Boulder, Colorado. SEPET (Spacecraft Electrical Performance and Evaluation Test) testing of the ERBE instruments at RCA's facilities in New Jersey was also extensively supported. Pre-launch testing of ERBS at Kennedy Space Center in Florida was carried out using the HP-1000 computer to acquire data from the Ball computer system on a real-time basis. Off-line data reduction was carried out to determine the status and performance of the ERBS instruments.

Data Reduction and Analysis

During the extensive testing of the three sets of ERBE instruments, several hundred magnetic tapes were filled with test data. Proper evaluation of these tests required the use of considerable software and computing facilities. TRW, the instrument fabricator, supplied a significant number of programs which were designed specifically to evaluate ERBE instrument test data. Under this contract, many additional programs were developed to augment the capabilities of the TRW software. These additional programs are described in the next section. Listings of some of these programs are included in the appendix. The vast majority of these programs were developed on an HP-1000 computer which was identical to the one used by TRW as the BCU (Bench

Check-Out) computer during testing.

SECTION 4 - SOFTWARE DEVELOPMENT

In the early stages of testing of the ERBE instruments at TRW, all the test data collected was reduced and analyzed on the same HP-1000 (BCU) computers used to run the tests and record test data. Although a fairly large set of computer programs was written by TRW to analyze the data on the HP-1000 computers, demand for time on the computers exceeded time available. Many of the tests performed on the ERBE instruments were run 24 hours a day for many days. It was felt that the data could be analyzed on NASA's large CDC NOS computers if the data tapes could be reformatted into a CDC compatible form. An HP-1000 program called MAGAN was developed under this contract to perform the reformatting and other utility functions. Originally, MAGAN was used by TRW to generate CDC compatible tapes which were then shipped to NASA for analysis. Later modifications to MAGAN added options to extract calibration data, to generate and update history tapes and to calculate time/cycle data. MAGAN could also be used to generate a comprehensive summary of the data on these tapes. On NASA's CDC - NOS computers a program called ERBDAT was designed to tabulate test data for both scanner and non-scanner instruments from these MAGAN reformatted data tapes.

Even though ERBDAT helped in the tabulation of ERBE test data, it soon became apparent that the use of an additional HP-1000 computer complete with the plotting and analysis software developed by TRW was needed. NASA/Langley acquired the use of an HP-1000 and work was begun on a system generation which would be

compatible with the systems on the three HP-1000 BCU computers at TRW. TRW's programs such as MTAPE, GNPLT, PLTMS, ERBDR, NSPLT were installed on the new HP-1000. Constant access to an HP-1000 made it possible to write additional programs which could augment those provided by TRW. CATLG was written to catalog the data records and SUMRY was written to supply time-listings of major frames, command echoes and status words. MAP was written to tabulate the timing of the ERBE data samples. INST was written to tabulate instrument data files extracted by the MTAPE program. Numerous improvements were made to the overall set of ERBE data reduction and analysis programs.

As the sets of ERBE instruments were readied for shipment to the satellite manufacturing facilities at RCA (ERBE/TIROS) and Ball Aerospace (ERBS), a new urgent need for software developed. During satellite integration many new test data tapes were produced. These data tapes were in totally different formats from the data tapes generated at TRW. In order to utilize the considerable software developed under this contract and by TRW, additional programs had to be developed to convert the data in these new formats back into a format usable by the existing software.

TIP was the first of these conversion programs written to convert TIP (Tiros Information Processor) data to TRW ERBE format. The TIP data was blocked in records of 1024 words. Of these 1024 words, only a small number of words were extracted to generate 10 minor frames of data. When 320 consecutive minor frames were assembled, a major frame was generated and written to

tape. Differences in the analog-to-digital converters in the TIROS satellite required changes in the data being written to the TRW formatted tapes.

PCM was a program designed to convert ERBS data tapes created at Ball Aerospace into TRW-PCM form (Pulse Code Modulated). A program called ALL was also written to generate TRW-PCM data tapes from ERBS All-Experiment formatted tapes.

Under some testing conditions at the satellite manufacturing facilities and at launch facilities, it was not always possible to have an HP-1000 computer handy. Utility programs were developed under this contract to run on portable Radio Shack TRS-80 Model 100 computers. These programs were useful in decoding command echoes, status words and in calculating position data. Program options also included capabilities to construct command sequences which could later be issued by the test conductors to the spacecraft.

APPENDIX

Page

PROGRAM LISTINGS AND USER NOTES:

MAGAN

MAGAN is used to analyze ERBE test data tapes using the HP-1000 BCU computer. To use MAGAN, simply mount a tape to be analyzed on either tape drive (LU 8 or 9) and then type MAGAN on the consol. MAGAN responds with a menu of options. Choose the desired option and proceed to answer any questions the program may ask.

OPTIONS

1. Tape to Disk, Reformat to New Tape

   Option one was designed to overcome the problem of reformatting a tape with only one tape drive. The tape is first copied to disk and then reformatted as it is written to the new tape. Reformatting in this case is to CDC-NOS format from TRW format. Tapes generated with this option can only be processed on CDC computers.

2. Tape to Tape With Reformatting

   This option is the same as option one except it takes advantage of a system with two tape drives.

3. Reformat and Write Disk to Tape

   After data has been copied to disk using some other option (such as option 1), this option will copy the data file from disk to a new tape. Multiple copies of the same

file can be created this way. The data is formatted on the tape in CDC format.

4. Catalog of Events

Mount a test data tape and this option will produce a listing of all the TRW test commands and comments recorded on the tape. Such a listing can prove to be very useful in locating the beginning of a calibration or test procedure.

5. Calculate Time/Cycle Data

This option is used to calculate "on time" and the number of cycles (on-off transitions) of the various "digital B" parameters. Digital B data is bilevel data which indicates which pulse load bus is in use, when the azimuth and elevation motors are on, etc.

6. Comprehensive Summary

This option generates a full report of the various types of commands, comments and error messages logged to a test data tape.

7. Change List Device

This enables a choice between printer or CRT (screen) for output of the program.

8. Rewind Tape

Program will rewind data tape.

9. Position Mag Tape Forward One Record

A useful option to position the tape beyond a trouble spot, such as an unexpected end-of-file.

10. Copy Disk to Tape With No Reformatting

This is used to make exact duplicates of test data tapes. Tapes generated with this option are usable only on the HP-1000 computer.

11. Enter Available Disk Cartridge Numbers

During tape copying to disk, considerable disk space is required. This option allows the user to select how much room on the disk he is willing to use to copy the tape.

12. Change Mag Tape Logical Unit

In a multiple tape unit system, this option allows either tape unit to be designated for use.

13. Update an Operating History Tape

After appropriate data has been selected for addition to the Operating History Tape, this option will append the records to the history tape.

14. List Contents of Operating History Tape

Lists the contents of the history tape to the printer.

15. Extract Cal Data and Update Cal History Tape

Extracts calibration data and updates Cal History Tape.

16. List Contents of Calibration History Tape

Provides hardcopy of Cal History Tape.

17. Exit MAGAN

```
$EMA(XYZ,0)
C
C
C
C     NAME:        MAGAN
C     MODULE:      PROGRAM
C     AUTHOR:      W. EDMONDS, NASA LANGLEY (804) 827-3761
C     REV DATE:    (830323.1611)
C
C     PURPOSE:
C     -------
C
C     MAGAN IS USED TO ANALYZE MAG TAPES PRODUCED WHEN TESTING
C     TIROS INSTRUMENT TAPES ON THE TRW BCU.  MAGAN PROVIDES
C     A MENU OF CAPABILITIES:
C
C    1 = Tape to Disk, Reformat to New Tape
C    2 = Tape to Tape, Reformat
C    3 = Reformat and Write Disk to Tape
C    4 = Catalog of Events
C    5 = Calculate Time/Cycle Info
C    6 = Provide Comprehensive Summmary of Tape
C    7 = Change List Device
C    8 = Rewind Mag Tape
C    9 = Position Mag Tape Forward by 1 Record
C    10 = Copy Disk to Tape as is (No Reformatting)
C    11 = Enter Available Disk Cartridge Numbers
C    12 = Change Mag Tape Logical Unit #
C    13 = Update an Operating History Tape
C    14 = List the Contents of an Operating History Tape
C    15 = Extract Cal Data from TRW Data Tape and Update Cal History Tape
C    16 = Provide hardcopy summary of contents of Calibration History Tape
C    17 = Exit
C
C    Enter a command:
C
C
C
      PROGRAM MAGAN( ),(830323.1611)
C
C
C
      INTEGER MAG0(3)                ! NAME OF 1ST SEGMENT
      INTEGER LBUF(100)              ! USED FOR LARGE BUFFER SUBROUTINE LGBUF
C
C
C
      COMMON /XYZ/ STVALU(2,8193)
      COMMON /BUF/ IDCB(144), IBUF(3938), IERR, LEN, IREC, HR,
     +             ICAR, NAM(3), ISIZ(2), LUL, IFIR, ILAS, JTYPE,
     +             ID, INST, MTFL, IVDF, ITEST(6,2), ICOND(3,2),
     +             NOSER(2,2), NOYR(2), NODAY(2), IREEL(2), NOHR(2),
     +             NOMIN(2), NOSEC(2), MOSEC(2), NCAR, IFCAR, LCAR,
     +             ISTAF(2), MTLU, LUNUM(2)
C
```

```
C
C
      COMMON /DIGB/ ISCOD(3,12), INCOD(3,12), ISDES(20,12),
     +              INDES(20,12), ONT(12,2),
     +              IDIG(2), ICCB(144), ICNAM(3), JSIZ(2),
     +              IDAY(12,2), JDAY
C
C
C
      DATA MAGO / 'MAGO  ' /
C
C
C
      LUT = LOGLU(IDUM)
      CALL LGBUF(LBUF,100)
      CALL SEGLD(MAGO,IRTN)
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 130 COMMON: (NONE)

```
C
C     ***************************************************************
C     ***************************************************************
C
      SUBROUTINE R6TIM(IBUF,ITIME,IRTN),READ THE ASCII TIME & CONVERT
C
C
C
      INTEGER IBUF(1), ITIME(1)
      INTEGER DAYMON(12), NAMMON(2,12)
C
C
C
      DATA DAYMON / 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31/
      DATA NAMMON / ' JAN', ' FEB', ' MAR', ' APR', ' MAY', ' JUN',
     +              ' JUL', ' AUG', ' SEP', ' OCT', ' NOV', ' DEC' /
C
C
C
      IRTN = 0
      IHR = 10*(IAND(177400B,IBUF(52))/256 - 60B) +
     +          IAND(377B,IBUF(52)) - 60B
      IF ((IBUF(58) .EQ. 2HPM) .AND. (IHR .NE. 12))  THEN
          IHR = IHR + 12
      ELSE IF ((IBUF(58) .EQ. 2HAM) .AND. (IHR .EQ. 12))  THEN
          IHR = 0
      ENDIF
      IF ((IHR .LT. 0) .OR. (IHR .GT. 23))  THEN
          IRTN = -1
          GOTO 1000
      ENDIF
      MINUT = 10*(IAND(377B,IBUF(53)) - 60B) +
     +         IAND(177400B,IBUF(54))/256 - 60B
      IF ((MINUT .LT. 0) .OR. (MINUT .GT. 59))  THEN
          IRTN = -1
          GOTO 1000
      ENDIF
      ISEC = 10*(IAND(177400B,IBUF(55))/256 - 60B) +
     +        IAND(377B,IBUF(55)) - 60B
      IF ((ISEC .LT. 0) .OR. (ISEC .GT. 59))  THEN
          IRTN = -1
          GOTO 1000
      ENDIF
C
C
C
      IDAYNM = 10*(IAND(377B,IBUF(62)) - 60B) +
     +             IAND(177400B,IBUF(63))/256 - 60B
      IF ((IDAYNM .LT. 1) .OR. (IDAYNM .GT. 31)) THEN
          IRTN = -1
          GOTO 1000
      ENDIF
      IYEAR = 1900 + 10*(IAND(177400B,IBUF(64))/256 - 60B) +
     +         IAND(377B,IBUF(64)) - 60B
      DO 100 I = 1,12
```

```
      IF ((IBUF(60) .EQ. NAMMON(1,I)) .AND.
     +    (IBUF(61) .EQ. NAMMON(2,I)))  THEN
         GOTO 120
      ENDIF
100   CONTINUE
120   MONTH = I
      IF (MONTH .EQ. 1)  THEN
          IDAY = IDAYNM
          GOTO 155
      ENDIF
      IDAY = 0
      DO 150 I = 1,MONTH-1
      IDAY = DAYMON(I) + IDAY
150   CONTINUE
      IF ((MOD(IYEAR,4) .EQ. 0) .AND. (MONTH .GT. 2))  THEN
          IDAY = IDAY + 1
      ENDIF
      IDAY = IDAY + IDAYNM
155   ITIME(1) = 0
      ITIME(2) = ISEC
      ITIME(3) = MINUT
      ITIME(4) = IHR
      ITIME(5) = IDAY
      ITIME(6) = IYEAR
1000  RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 380 COMMON: (NONE)

```
155 C
156 C     ************************************************************
157 C     ************************************************************
158 C
159       SUBROUTINE SUBTM(CURTM,SBTIM,NUTIM,IRTN),REV NC 11-19-81 BY DHL
160       INTEGER CURTM(6), SBTIM(6), NUTIM(6), IMAX(6)
161       INTEGER DMTIM(6)
162 C
163 C
164 C
165       DATA IMAX/100, 60, 60, 24, 365, 0/
166 C
167 C
168 C
169       DO 100 I = 1,6
170       DMTIM(I) = CURTM(I)
171 100   CONTINUE
172       DO 500 I = 1,6
173       NUTIM(I) = DMTIM(I) - SBTIM(I)
174       IF (NUTIM(I) .GE. 0)   GO TO 500
175 C
176 C     MUST BORROW - PERHAPS MORE THAN ONCE.
177 C
178       NUTIM(I) = NUTIM(I) + IMAX(I)
179       DO 300 J = I+1,6
180       IF (J .EQ. 6)  GOTO 600
181       DMTIM(J) = DMTIM(J) - 1
182       IF (DMTIM(J) .GE. 0)   GO TO 500
183       DMTIM(J) = DMTIM(J) + IMAX(J)
184   300 CONTINUE
185   500 CONTINUE
186       DO 550 N = 1,6
187   550 IF (NUTIM(N) .LT. 0)  GOTO 600
188       IRTN = 0
189       RETURN
190 C
191 C     ERROR - CAN'T SUBTRACT GREATER TIME FROM LESSER TIME
192 C
193   600 IRTN = -1
194       RETURN
195       END
```

```
FTN4X COMPILER: HP92834 REV.2130 (810716)

**  NO WARNINGS **  NO ERRORS **   PROGRAM:   143       COMMON: (NONE)
```

```
196 C
197 C     ***************************************************************
198 C     ***************************************************************
199 C
200       SUBROUTINE HOURS(ITIME,EHOUR)
201 C
202 C
203 C
204       INTEGER ITIME(1)
205 C
206 C
207 C
208       EHOUR = ITIME(4) + ITIME(5)*24. + ITIME(3)/60. + ITIME(2)/3600.
209       RETURN
210       END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 65 COMMON: (NONE)

```
C
C     ***************************************************************
C     ***************************************************************
C
      SUBROUTINE PUTTM(FRTIM1,TOTIM2)
C
C
C
      INTEGER FRTIM1(1), TOTIM2(1)
C
C
C
      DO 100 I = 1,6
      TOTIM2(I) = FRTIM1(I)
  100 CONTINUE
      RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 31 COMMON: (NONE)

```
228 C
229 C     ************************************************************
230 C     ************************************************************
231 C
232       SUBROUTINE COPI(KBUF)
233 C
234 C
235 C
236       COMMON /BUF/ IDCB(144), IBUF(3938), IERR, LEN, IREC, HR,
237      +             ICAR, NAM(3), ISIZ(2), LUL, IFIR, ILAS, JTYPE,
238      +             ID, INST, MTFL, IVDF, ITEST(6,2), ICOND(3,2),
239      +             NOSER(2,2), NOYR(2), NODAY(2), IREEL(2), NOHR(2),
240      +             NOMIN(2), NOSEC(2), MOSEC(2), NCAR, IFCAR, LCAR,
241      +             ISTAF(2), MTLU, LUNUM(2)
242 C
243 C
244 C
245       DIMENSION IDC(144), KFORM(4)
246      +,KBUF(2400),IFOR(7),IFIV(4),NFORM(4),MFORM(2)
247 C
248 C
249 C
250       DATA IFOR / '(I3,I3,E14.8) ' /
251       DATA IFIV / '(3E12.6)' /
252       DATA NFORM / '(48006 )' /
253       DATA KFORM / '(80006 )' /
254       DATA MFORM / '(I6)' /
255 C
256 C
257 C
258       LUT = LOGLU(IDUM)
259       WRITE(LUL,9100)
260       MTALT = 0
261       CALL OPENF(IDC,IERR,MTLU)
262       IF (IERR .LT. 0) THEN
263    25     WRITE(LUT,9110) IERR
264           GOTO 5000
265       ENDIF
266       IFC = IFCAR
267       LC = LCAR
268    35 WRITE(LUT,9120)
269       READ(LUT,9122) IANS
270       IF ((IANS .NE. 2HYE) .AND. (IANS .NE. 2HNO)) GOTO 35
271       IF (IANS .EQ. 2HYE) THEN
272    40     WRITE(LUT,9125)
273           READ(LUT,*) IDUM
274           IF ((IDUM .LT. 1) .OR. (IDUM .GT. 63)) GOTO 40
275           MTALT = IDUM
276           LC = IFCAR
277       ELSE
278           GOTO 50
279       ENDIF
280 C
281 C     *****  WHEN MTALT IS NOT 0, THEN THIS IS TAPE TO TAPE COPY.
282 C     *****  THIS INDICATES THAT THE DO 10 LOOP BELOW WILL ONLY
```

```
283 C     *****  NEED TO BE EXECUTED ONCE. THEREFORE LC = IFC = IFCAR.
284 C     *****  WHEN MTALT = 0, THEN THIS IS COPY FROM DISK TO TAPE.
285 C     *****  SINCE THE DISK FILE MAY OCCUPY MORE THAN ONE CARTRIDGE,
286 C     *****  IT IS NECESSARY TO LOOP THROUGH THE CARTRIDGES.
287 C
288       CALL OPENF(IDCB,IERR,MTALT)
289       IF (IERR .LT. 0)  THEN
290           WRITE(LUT,9130) IERR
291           GOTO 5000
292       ENDIF
293       LINE = 0
294    50 DO 1000 JCAR = IFC,LC
295       IF (MTALT .EQ. 0) CALL OPEN(IDCB,IERR,NAM,0,0,JCAR)
296       IF (IERR .LT. 0)  GOTO 25
297    65 CALL READS
298       IF (LEN .LE. 0)  GOTO 550
299       IF (IERR .LT. 0)  THEN
300           WRITE(LUT,9140) IERR, LEN
301           GOTO 550
302       ENDIF
303       IF (IBUF .EQ. 0)  GOTO 4000
304 C
305 C     *****  RECORD TYPE 0 INDICATES END OF INFORMATION.
306 C
307       IF (IBUF .EQ. 1)  THEN
308 C
309 C     *****  INSTRUMENT DATA.
310 C
311       IF (IVDF .EQ. 0)  GOTO 65
312 C
313 C     *****  THE VALID DATA FLAG IS SET TO ZERO FOR ALL INCOMPLETE
314 C     *****  MAJOR FRAMES BY SUBROUTINE READS.
315 C
316       CALL HEADR(KBUF(2),1)
317       LINE = LINE + 1
318       IF (LINE .GT. 52)  THEN
319           LINE = 0
320           WRITE(LUL,9100)
321       ENDIF
322       IVDF = IBUF(IND(2,7))
323       IF (IVDF .NE. 1)  GOTO 65
324       ENCODE(2,9150,KBUF) IBUF(1)
325       CALL ENC(6,KBUF(67),LEN,1,1,MFORM,2)
326 C
327 C
328 C
329       DO 180 LLL = 68,70
330       KBUF(LLL) = 2H
331   180 CONTINUE
332 C
333 C     *****  THE LAST 3 STATEMENTS WERE ADDED TO FILL ANY EVEN MULTIPLE
334 C     *****  OF 10 LOCATIONS IN KBUF. EVERY 10 LOCATIONS IN HP YIELDS
335 C     *****  2 WORDS ON CDC MACHINES. OK?
336 C
337       CALL WRITF(IDC,IERR,KBUF,70)
```

```
338         IF (IERR .EQ. -12) CALL NTAPE(IDC,IERR,KBUF,70,8)
339         IF (IERR .LT. 0)  GOTO 25
340         IF (IBUF(2) .EQ. 0)  GOTO 300
341 C
342 C       *****  GOTO 300 TO PROCESS NON SCANNER
343 C
344         ILEN = LEN - 479
345         IF (LEN .NE. 3938)  GOTO 65
346         DO 200 N = 99,ILEN,480
347         M = N + 479
348         CALL ENC(2880,KBUF,IBUF,N,M,NFORM,4)
349         CALL WRITF(IDC,IERR,KBUF,1440)
350         IF (IERR .EQ. -12) CALL NTAPE(IDC,IERR,KBUF,1440,8)
351         IF (IERR .LT. 0)  GOTO 25
352     200 CONTINUE
353         GOTO 65
354     300 CONTINUE
355         ILEN = LEN - 799
356         IF (LEN .NE. 1642)  GOTO 65
357         DO 400 N = 43,ILEN,800
358         M = N + 799
359         CALL ENC(4800,KBUF,IBUF,N,M,KFORM,4)
360         CALL WRITF(IDC,IERR,KBUF,2400)
361         IF (IERR .EQ. -12) CALL NTAPE(IDC,IERR,KBUF,2400,8)
362         IF (IERR .LT. 0)  GOTO 25
363     400 CONTINUE
364         GOTO 65
365 C
366 C
367 C
368         ELSE IF (IBUF .EQ. 2)  THEN
369 C
370 C       *****  PROCESS AMON DATA HERE.
371 C
372         NC = IBUF(8)
373 C
374 C       *****  GET # OF CHANNELS IN THIS RECORD OF AMON DATA
375 C
376         ENCODE(30,9180,KBUF) (IBUF(I),I = 1,8)
377         DO 500 J = 1,NC
378         L = 9 + (J-1)*4
379         K = 16 + (J-1)*10
380         M = L + 3
381         CALL ENC(20,KBUF(K),IBUF,L,M,IFOR,7)
382     500 CONTINUE
383         K = NC*10 + 16
384         L = L + 4
385         M = L + 18
386         CALL ENC(36,KBUF(K),IBUF,L,M,IFIV,4)
387         CALL WRITF(IDC,IERR,KBUF,K+17)
388         IF (IERR .EQ. -12) CALL NTAPE(IDC,IERR,KBUF,K+17,8)
389         IF (IERR .LT. 0)  GOTO 25
390         GOTO 65
391 C
392 C
```

```
C
      ELSE
C
C     *****  PROCESS ALL RECORD TYPES EXCEPT AMON & DIG A
C
      ENCODE(2,9150,ICODE) IBUF(1)
      IBUF(1) = ICODE
      CALL WRITF(IDC,IERR,IBUF,LEN)
      IF (IERR .EQ. -12) CALL NTAPE(IDC,IERR,IBUF,LEN,8)
      IF (IERR .LT. 0)  GOTO 25
      GOTO 65
      ENDIF
C
C
C
  550 IF (MTALT .NE. 0) CALL RWNDF(IDCB)
      CALL CLOSE(IDCB)
 1000 CONTINUE
 4000 CALL EXEC(3,MTLU+100B)
      CALL EXEC(3,MTLU+100B)
      CALL RWNDF(IDC)
C
C
C
 5000 RETURN
C
C     *****  FORMAT STATEMENTS
C
 9100 FORMAT("1",/,T2,"TEST",T14,"TEST",T22,"SERIAL",T57,"DECIMAL",
     +       T65,"MAJ",T72,"TAPE",T80,"REEL",T85,"IN",T90,"DATA",
     +       T95,"INSTR",T101,"STN",/,
     +       T2,"PROCEDURE",T14,"COND.",T22,"NUMBER",T32,"TIME",
     +       T47,"DATE",T57,"TIME",T65,"FR #",T72,"REC #",T80,"#",
     +       T85,"VAC",T90,"OK",T95,"TYPE",T101,"ID",T107,"DIST",
     +       T116,"AZ POS",T126,"EL POS",2/)
 9110 FORMAT(/,"FMGR ERROR # ",I4," ON MAG TAPE OPERATION")
 9120 FORMAT(/,"Do you want to Copy from the Source Tape Unit to ",/,
     +         "the Destination Tape Unit? (YES or NO): _")
 9122 FORMAT(1A2)
 9125 FORMAT(/,"Enter the Logical Unit # of the Source Tape Unit: _")
 9130 FORMAT(/"FMGR ERROR # ",I4," ON DESTINATION TAPE UNIT")
 9140 FORMAT(/,"FMGR ERROR # ",I4," RECORD LENGTH = ",I5)
 9150 FORMAT(I2)
 9180 FORMAT(I2,6I4,I2,2X)
      END


FTN4X COMPILER: HP92834 REV.2130 (810716)

**  NO WARNINGS **  NO ERRORS **   PROGRAM:  1152       COMMON: (NONE)
```

```
C
C     ****************************************************************
C     ****************************************************************
C
      SUBROUTINE NTAPE(IDC,IERR,KBUF,LEN,LUL)
C
C
C
      DIMENSION IDC(144), KBUF(2400)
C
C
C
      LUT = LOGLU(IDUM)
      WRITE(LUT,9100)
      CALL RWNDF(IDC)
      CALL CLOSE(IDC)
  100 WRITE(LUT,9120)
      READ(LUT,9150) IANS
      IF (IANS .EQ. 2HGO)  GOTO 200
      WRITE(LUT,9140)
      READ(LUT,9150) IANS
      IF (IANS .EQ. 2HYE)  THEN
          CALL EXEC(6)
      ELSE
          GOTO 100
      ENDIF
  200 CALL OPENF(IDC,IERR,LUL)
      CALL WRITF(IDC,IERR,KBUF,LEN)
      RETURN
C
C     *****  FORMAT STATEMENTS
C
 9100 FORMAT(/,"End-of-Tape on Mag Tape  -  after Mag Tape rewinds,",/,
     +         "Mount Continuation Tape!")
 9120 FORMAT(/,"Enter GO when ready: _")
 9140 FORMAT(/,"Do you want to abort? (YES or NO): _")
 9150 FORMAT(1A2)
      END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:   193       COMMON: (NONE)
```

```
476 C
477 C     ************************************************************
478 C     ************************************************************
479 C
480       SUBROUTINE MSTOR(IDC,IP)
481 C
482 C
483 C
484       COMMON /BUF/ IDCB(144), IBUF(3938), IERR, LEN, IREC, HR,
485      +             ICAR, NAM(3), ISIZ(2), LUL, IFIR, ILAS, JTYPE,
486      +             ID, INST, MTFL, IVDF, ITEST(6,2), ICOND(3,2),
487      +             NOSER(2,2), NOYR(2), NODAY(2), IREEL(2), NOHR(2),
488      +             NOMIN(2), NOSEC(2), MOSEC(2), NCAR, IFCAR, LCAR,
489      +             ISTAF(2), MTLU, LUNUM(2)
490 C
491 C
492 C
493       INTEGER*4 ITLEN, IBLEN
494 C
495 C
496 C
497       DIMENSION IP(8), IDC(144)
498 C
499 C
500 C
501       DATA ITYP/3/
502 C
503 C
504 C
505       LUT = LOGLU(IDUM)
506       IREC = 0
507       NCAR = 0
508 C
509 C     *****  NCAR COUNTS THE NUMBER OF CARTRIDGES USED TO HOLD DATA FILE.
510 C
511       JERR = 0
512       ITLEN = 0
513       LINE = 0
514       IBLEN = 0
515       WRITE(LUL,9100)
516       CALL OPENF(IDC,IERR,MTLU)
517       IF (IERR .LT. 0)  THEN
518           WRITE(LUT,9110) IERR
519           GOTO 5000
520       ENDIF
521 C
522 C     *****  FOR EACH CARTRIDGE . . .
523 C
524       DO 1000 IC = IFCAR,LCAR
525       CALL PURGE(IDCB,IERR,NAM,0,IC)
526       CALL CREAT(IDCB,IERR,NAM,ISIZ,ITYP,0,IC)
527       IF (IERR .LT. 0)  THEN
528           WRITE(LUT,9120) IERR
529           GOTO 5000
530       ENDIF
```

```
      NCAR = NCAR + 1
  125 IF (JERR .NE. -33) CALL READF(IDC,IERR,IBUF,3938,ILEN)
      IF (IERR .EQ. -5)  GOTO 125
      LEN = ILEN
      IF ((ILEN .LT. 0) .OR. (IERR .LT. 0))  GOTO 1200
      CALL WRITF(IDCB,JERR,IBUF,ILEN)
      IF (JERR .EQ. -33)  GOTO 200
      IF (JERR .LT. 0)  THEN
          WRITE(LUT,9130) JERR
          GOTO 1200
      ENDIF
      IF (ILEN .LT. 0)  GOTO 1200
      ITLEN = ITLEN + ILEN
      IF (ILEN .GT. 200) IBLEN = IBLEN + ILEN
      IREC = IREC + 1
      LEN = ILEN
      IF (IBUF .EQ. 1)  THEN
          HR = TIME(IBUF(3))
          CALL HEADR(IP,0)
          LINE = LINE + 1
          GOTO 175
      ENDIF
      IF (IP(IBUF) .EQ. 1)  THEN
          WRITE(LUL,9140) (IBUF(K),K = 2,LEN)
          LINE = LINE + 1
      ENDIF
      IF (LINE .GT. 52)  THEN
          LINE = 0
          WRITE(LUL,9100)
      ENDIF
  175 IF (JERR .GE. 0)  GOTO 125
      GOTO 1100
  200 CALL CLOSE(IDCB)
 1000 CONTINUE
      WRITE(LUT,9150)
      GOTO 1300
 1100 WRITE(LUT,9110) JERR
 1200 IBUF(1) = 0
      ILEN = 1
      CALL WRITF(IDCB,IERR,IBUF,ILEN)
      IF (IERR .LT. 0) WRITE(LUT,9160) IERR
      CALL CLOSE(IDCB)
 1300 WRITE(LUT,9170) IREC, ITLEN, IBLEN
      WRITE(LUL,9170) IREC, ITLEN, IBLEN
      CALL RWNDF(IDC)
      CALL CLOSE(IDC)
 5000 RETURN
C
C     *****  FORMAT STATEMENTS
C
 9100 FORMAT("1",/,T2,"TEST",T14,"TEST",T22,"SERIAL",T57,"DECIMAL",
     +       T65,"MAJ",T72,"TAPE",T80,"REEL",T85,"IN",T90,"DATA",
     +       T95,"INSTR",T101,"STN",/,
     +       T2,"PROCEDURE",T14,"COND.",T22,"NUMBER",T32,"TIME",
     +       T47,"DATE",T57,"TIME",T65,"FR #",T72,"REC #",T80,"#",
```

```
586       +        T85,"VAC",T90,"OK",T95,"TYPE",T101,"ID",T107,"DIST",
587       +        T116,"AZ POS",T126,"EL POS",2/)
588    9110 FORMAT(/,"FMGR ERROR # ",I4)
589    9120 FORMAT(/,"FMGR ERROR # ",I4," ATTEMPTING TO OPEN OR CREATE ",
590       +          "MTDFIL")
591    9130 FORMAT(/,"FMGR ERROR # ",I4," IN WRITING TO DLU ",I2)
592    9140 FORMAT(1X,60A2)
593    9150 FORMAT(/,"NOT ENOUGH ROOM ON DISK FOR ALL DATA FROM TAPE!")
594    9160 FORMAT(/,"FMGR ERROR # ",I4," ON DISK FILE")
595    9170 FORMAT(/,1X,"NUMBER OF RECORDS: ",I5,/,
596       +          1X,"TOTAL NUMBER OF WORDS: ",I10,/,
597       +          1X,"TOTAL NUMBER OF BINARY WORDS: ",I10)
598         END


FTN4X COMPILER: HP92834 REV.2130 (810716)


 **  NO WARNINGS **  NO ERRORS **   PROGRAM:    705        COMMON: (NONE)
```

```
599 C
600 C     ***************************************************************
601 C     ***************************************************************
602 C
603       BLOCK DATA DIGB
604 C
605 C
606 C
607       COMMON /DIGB/ ISCOD(3,12), INCOD(3,12), ISDES(20,12),
608      +              INDES(20,12), ONT(12,2),
609      +              IDIG(2), ICCB(144), ICNAM(3), JSIZ(2),
610      +              IDAY(12,2), JDAY
611 C
612 C
613 C
614       DATA ICNAM / 'CYCTIM' /
615       DATA JSIZ /100,0/
616       DATA ISCOD / 'SDB1  SDB2  SDB3  SDB4  SDB5                ',
617      +             'SDB6  SDB9  SDB10 SDB7  SDB8  ' /
618       DATA INCOD / 'NDB1  NDB2  NDB3  NDB4  NDB5                ',
619      +             'NDB6  NDB9  NDB10 NDB7  NDB8  ' /
620       DATA ISDES /240*2H  /
621       DATA INDES /240*2H  /
622       DATA ISDES / 'INSTRUMENT POWER                           ',
623      +             'PULSE LOAD BUS A                           ',
624      +             'PULSE LOAD BUS B                           ',
625      +             'HEAD STANDBY HEATER POWER                  ',
626      +             'DIGITAL B SPARE                            ',
627      +             'NOT USED                                   ',
628      +             'NOT USED                                   ',
629      +             'BLACKBODY HEATER POWER                     ',
630      +             'AZIMUTH MOTOR POWER                        ',
631      +             'SCAN MOTOR POWER                           ',
632      +             'PED STANDBY HEATER POWER                   ',
633      +             'DIGITAL B SPARE                            ' /
634       DATA INDES / 'INSTRUMENT POWER                           ',
635      +             'PULSE LOAD BUS A                           ',
636      +             'PULSE LOAD BUS B                           ',
637      +             'HEAD STANDBY HEATER POWER                  ',
638      +             'INSTRUMENT HEATER BUS POWER                ',
639      +             'NOT USED                                   ',
640      +             'NOT USED                                   ',
641      +             'BLACKBODY HEATER BUS POWER                 ',
642      +             'AZIMUTH MOTOR POWER                        ',
643      +             'ELEVATION MOTOR POWER                      ',
644      +             'PED STANDBY HEATER POWER                   ',
645      +             'DIGITAL B SPARE                            ' /
646       END

FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM: (NONE)      COMMON: (NONE)

BLOCK COMMON DIGB     SIZE:   776
```

```
C
C     ******************************************************
C     ******************************************************
C
      BLOCK DATA BUF
C
C
C
      COMMON /BUF/ IDCB(144), IBUF(3938), IERR, LEN, IREC, HR,
     +             ICAR, NAM(3), ISIZ(2), LUL, IFIR, ILAS, JTYPE,
     +             ID, INST, MTFL, IVDF, ITEST(6,2), ICOND(3,2),
     +             NOSER(2,2), NOYR(2), NODAY(2), IREEL(2), NOHR(2),
     +             NOMIN(2), NOSEC(2), MOSEC(2), NCAR, IFCAR, LCAR,
     +             ISTAF(2), MTLU, LUNUM(2)
C
C
C
      DATA IFIR, ILAS /1,32000/
      DATA ITEST /12*2H  /
      DATA ICOND /6*2H  /
      DATA NOSER /4*2H  /
      DATA NOYR /2*0/
      DATA NODAY /2*0/
      DATA IREEL /2*0   /
      DATA NOHR /2*0/
      DATA NOMIN /2*0/
      DATA NOSEC /2*0/
      DATA MOSEC /2*0/
      DATA ISTAF /-1,-1/
      DATA IREC /0/
      DATA ISIZ /-1,0/
      DATA NAM / 'MTDFIL' /
      DATA ICAR /20/
      DATA IFCAR, LCAR /20,23/
      DATA NCAR /1/
      DATA LUL /6/
      DATA MTLU /8/
      DATA LUNUM /6,8/
      END

FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM: (NONE)      COMMON: (NONE)

BLOCK COMMON BUF      SIZE:  4145
```

```
686 C
687 C     ************************************************************
688 C     ************************************************************
689 C
690       SUBROUTINE READS
691 C
692 C
693 C
694       COMMON /BUF/ IDCB(144), IBUF(3938), IERR, LEN, IREC, HR,
695      +             ICAR, NAM(3), ISIZ(2), LUL, IFIR, ILAS, JTYPE,
696      +             ID, INST, MTFL, IVDF, ITEST(6,2), ICOND(3,2),
697      +             NOSER(2,2), NOYR(2), NODAY(2), IREEL(2), NOHR(2),
698      +             NOMIN(2), NOSEC(2), NOSEC(2), NCAR, IFCAR, LCAR,
699      +             ISTAF(2), MTLU, LUNUM(2)
700 C
701 C
702 C
703       LUT = LOGLU(IDUM)
704       CALL READF(IDCB,IERR,IBUF,3938,LEN)
705       IF (IERR .LT. 0)  GOTO 5000
706       JTYPE = IBUF
707       IREC = IREC + 1
708 C
709 C     *****  IF THIS IS NOT A RECORD OF INSTRUMENT DATA, RETURN.
710 C
711       IF (JTYPE .NE. 1)  GOTO 5000
712 C
713 C     *****  CHECK THE DATA VALID FLAG.
714 C     *****  IF DATA VALID FLAG IS NOT 1, RETURN.
715 C
716       IVDF = IBUF(IND(2,7))
717       IF (IVDF .NE. 1)  GOTO 5000
718       INST = IBUF(2)
719       IF (((INST .EQ. 0) .AND. (LEN .NE. 1642)) .OR.
720      +    ((INST .EQ. 1) .AND. (LEN .NE. 3938)))  THEN
721           GOTO 5000
722       ENDIF
723       IDSTA = IBUF(IND(3,4))
724       IF ((IDSTA .EQ. 2HA ) .OR. (IDSTA .EQ. 0))  THEN
725           ID = 0
726       ELSE
727           ID = 1
728       ENDIF
729       HR = TIME(IBUF(3))
730       II = ID + 1
731       IF (ISTAF(II) .GT. -1)  GOTO 5000
732       ISTAF(II) = INST
733       ITEST(1,II) = IBUF(IND(5,2))
734 C
735 C
736 C
737       DO 50 LLL = 1,5
738       ITEST(LLL+1,II) = IBUF(IND(LLL,3))
739    50 CONTINUE
740 C
```

```
C
C
      DO 75 LLL = 1,3
      ICOND(LLL,II) = IBUF(IND(LLL,5))
   75 CONTINUE
C
C
C
      NOSER(1,II) = IBUF(IND(1,6))
      NOSER(2,II) = IBUF(IND(2,6))
      NOYR(II) = IBUF(IND(1,2))
      NODAY(II) = IBUF(IND(5,1))
      NOHR(II) = IBUF(IND(4,1))
      NOMIN(II) = IBUF(IND(3,1))
      NOSEC(II) = IBUF(IND(2,1))
      MOSEC(II) = IBUF(IND(1,1))
      IREEL(II) = IBUF(IND(4,5))
C
C
C
      WRITE(LUT,9100) IDSTA
100   WRITE(LUT,9120) (ITEST(J,II),J = 1,6),(ICOND(J,II),J = 1,3),
     +(NOSER(J,II),J = 1,2),IREEL(II)
      READ(LUT,*) ICODE
      IF (ICODE .EQ. 0)  GOTO 5000
      GOTO (10,20,30,40) ICODE
10    WRITE(LUT,9140)
      READ(LUT,9180) (ITEST(J,II),J = 1,6)
      GOTO 100
20    WRITE(LUT,9160)
      READ(LUT,9200) (ICOND(J,II),J = 1,3)
      GOTO 100
30    WRITE(LUT,9220)
      READ(LUT,9240) (NOSER(J,II),J = 1,2)
      GOTO 100
40    WRITE(LUT,9260)
      READ(LUT,*) IREEL(II)
      GOTO 100
 5000 RETURN
C
C     *****  FORMAT STATEMENTS
C
 9100 FORMAT(/,"Station ",1A2," Header Information: ")
 9120 FORMAT(2/,"Code     Description          Value",2/,
     +          " 1       Test Procedure Name  ",6A2,/,
     +          " 2       Test Conductor       ",3A2,/,
     +          " 3       Instrument S.N.      ",2A2,/,
     +          " 4       Mag Tape Reel #    ",I3,2/,
     +          "Enter the code to modify (0 for no change): _")
 9140 FORMAT(/,"Enter Test Procedure Name (12 chars max): _")
 9160 FORMAT(/,"Enter Test Conductor (6 chars max): _")
 9180 FORMAT(6A2)
 9200 FORMAT(3A2)
 9220 FORMAT(/,"Enter Instrument Serial Number (4 chars max): _")
 9240 FORMAT(2A2)
```

```
  796  9260 FORMAT(/,"Enter Mag Tape Reel Number: _")
  797       END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:    769        COMMON: (NONE)
```

```
798 C
799 C     ***************************************************************
800 C     ***************************************************************
801 C
802       FUNCTION TIME(IB)
803       DIMENSION IB(4)
804       TIME = FLOAT(IB(4)) + FLOAT(IB(3))/60. + FLOAT(IB(2))/3600.
805      + + FLOAT(IB(1))/360000.
806       RETURN
807       END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 62 COMMON: (NONE)

```
808 C
809 C     ************************************************************
810 C     ************************************************************
811 C
812       SUBROUTINE HEADR(KBUF,IK)
813 C
814 C
815 C
816       COMMON /BUF/ IDCB(144), IBUF(3938), IERR, LEN, IREC, HR,
817      +             ICAR, NAM(3), ISIZ(2), LUL, IFIR, ILAS, JTYPE,
818      +             ID, INST, MTFL, IVDF, ITEST(6,2), ICOND(3,2),
819      +             NOSER(2,2), NOYR(2), NODAY(2), IREEL(2), NOHR(2),
820      +             NOMIN(2), NOSEC(2), MOSEC(2), NCAR, IFCAR, LCAR,
821      +             ISTAF(2), MTLU, LUNUM(2)
822 C
823 C     *****  PROCESSING OF DIGITAL A DATA (MAJOR FRAME HEADER INFO)
824 C
825       DIMENSION KBUF(128), KTIME(6), ITBUF(13)
826       DIMENSION IDIST(2), IAZ(2), IEL(2)
827 C
828 C
829 C
830       EQUIVALENCE (IDIST(1),DIST)
831       EQUIVALENCE (IEL(1),EL)
832       EQUIVALENCE (IAZ(1),AZ)
833 C
834 C
835 C
836       LUT = LOGLU(IDUM)
837       MAFR = IBUF(IND(2,2))
838       KTIME(6) = IBUF(IND(1,2))
839       DO 50 K = 1,5
840       KTIME(K) = IBUF(IND(K,1))
841    50 CONTINUE
842       HR = TIME(IBUF(3))
843       MIFR = IBUF(IND(1,4))
844       IDINS = IBUF(2)
845       IDSTA = IBUF(IND(3,4))
846       IF ((IDSTA .EQ. 2HA ) .OR. (IDSTA .EQ. 0))  THEN
847           ID = 0
848           ITSTA = 2HA
849       ELSE
850           ID = 1
851           ITSTA = 2HB
852       ENDIF
853 C
854 C
855 C
856       IF (IDINS .EQ. 0)  THEN
857           ITINS = 2HNS
858       ELSE
859           ITINS = 2HSC
860       ENDIF
861       IDSTA = ID
862       IDD = ID + 1
```

```
      MAFT = IBUF(IND(4,4))
      MAGF = IBUF(IND(5,4))
      NTAB = IBUF(IND(5,5))
      IVA = IBUF(IND(3,6))
      IF (IVA .EQ. 0)  THEN
          IVANS = 2HN
      ELSE
          IVANS = 2HY
      ENDIF
      LNPF = IBUF(IND(1,7))
      NVALF = IBUF(IND(2,7))
      IF (NVALF .EQ. 0)  THEN
          INANS = 2HN
      ELSE
          INANS = 2HY
      ENDIF
C     IREEL(IDD) = IBUF(IND(4,5))
C
C
C
C     NOSER(1,IDD) = IBUF(IND(1,6))
C     NOSER(2,IDD) = IBUF(IND(2,6))
C
C
C
C     ITEST(1,IDD) = IBUF(IND(5,2))
C     DO 75 LLL = 1,5
C     ITEST(LLL+1,IDD) = IBUF(IND(LLL,3))
C  75 CONTINUE
C
C
C
      IDIST(1) = IBUF(IND(3,2))
      IDIST(2) = IBUF(IND(4,2))
      IAZ(1) = IBUF(IND(4,6))
      IAZ(2) = IBUF(IND(5,6))
      IEL(1) = IBUF(IND(4,7))
      IEL(2) = IBUF(IND(5,7))
      CALL CNVTM(KTIME,ITBUF)
      WRITE(LUL,1) (ITEST(J,IDD),J = 1,6),(ICOND(J,IDD),J = 1,3),
     +(NOSER(J,IDD),J = 1,2),ITBUF,
     +HR,MAFR,IREC,IREEL(IDD),IVANS,INANS,ITINS,ITSTA,DIST,AZ,EL
1     FORMAT(1X,6A2,1X,3A2,1X,2A2,2X,13A2,1X,F8.4,1X,
     +I5,1X,I5,4X,I3,4X,A2,3X,A2,3X,A2,4X,A2,3(E9.3,1X))
      IF (IK .NE. 0) ENCODE(133,1,KBUF) (ITEST(J,IDD),J = 1,6),
     +(ICOND(J,IDD),J = 1,3),(NOSER(J,IDD),J = 1,2), ITBUF,
     +     HR,MAFR,IREC,IREEL(IDD),IVANS,INANS,ITINS,ITSTA,DIST,
     +AZ,EL
C
C
C
C
      RETURN
      END
```

```
FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:    502        COMMON: (NONE)
```

```
C
C     ***************************************************
C     ***************************************************
C
      FUNCTION IND(I,J)
C
C
C
      COMMON /BUF/ IDCB(144), IBUF(3938), IERR, LEN, IREC, HR,
     +             ICAR, NAM(3), ISIZ(2), LUL, IFIR, ILAS, JTYPE,
     +             ID, INST, MTFL, IVDF, ITEST(6,2), ICOND(3,2),
     +             NOSER(2,2), NOYR(2), NODAY(2), IREEL(2), NOHR(2),
     +             NOMIN(2), NOSEC(2), MOSEC(2), NCAR, IFCAR, LCAR,
     +             ISTAF(2), MTLU, LUNUM(2)
C
C
C
      INT = 12
      IF (IBUF(2) .EQ. 0) INT = 5
      IND = (J-1)*INT + I + 2
      RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 26 COMMON: (NONE)

```
939 C
940 C     ************************************************************
941 C     ************************************************************
942 C
943       SUBROUTINE ENC(N,KB,K,J,M,IFORM,L)
944       DIMENSION K(480),KB(480),IFORM(L)
945       ENCODE(N,IFORM,KB)(K(I),I=J,M)
946       RETURN
947       END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 63 COMMON: (NONE)

SUMRY

SUMRY is a small but useful utility which provides a rapid means of listing the major frame headers contained on a test data tape. It lists the time of each major frame as well as the command echo and instrument status. SUMRY will work with either TRW tapes in "ERBE" format (for the TIROS instruments) or with tapes in the "PCM" format (for ERBS instruments).

To use this program, mount the desired tape and type SUMRY on the consol. Answer the prompts as required.

```
$FILES(1,0)
      PROGRAM SUMRY
C
C
C
C         SUMRY IS A PROGRAM USED TO SUMMARIZE ERBE DATA TAPES.
C         IT READS "TRW FORMAT" TAPES (EITHER TIROS "ERBE" FORM OR ERBS
C         "PCM" FORM) AND GENERATES LISTINGS OF MAJOR FRAME HEADER DATA.
C
C
C
      COMMON /BUF/ INST,IANA
      DIMENSION LINE(65),IDESC(30),ITAPE(10)
      LUT=LOGLU(IDUM)          ! GET LU OF TERMINAL BEING USED
100   WRITE(LUT,10)
10    FORMAT(" ENTER TAPE NUMBER (TRW# E.G.: E1147N)")
      READ(LUT,11)ITAPE        ! GET TAPE NUMBER FOR PRINTOUT
11    FORMAT(10A2)
      WRITE(LUT,12)
12    FORMAT(" ENTER TEST DESCRIPTION (UP TO 60 CHARACTERS)")
      READ(LUT,13)IDESC        ! GET TEST DESCRIPTION FOR PRINTOUT
13    FORMAT(30A2)
      WRITE(6,14)ITAPE,IDESC   ! WRITEOUT TAPE# AND TEST DESCRIPTION
14    FORMAT(1H1,1X,10A2,/,1X,30A2,    )
      WRITE(6,15)
15    FORMAT(///)
      OPEN(8)                    ! OPEN THE TAPE FILE
      IREC=0
1     READ(8,ERR=98,END=99)ITYPE,INST
C         READ A RECORD TO DETERMINE ITS RECORD TYPE AND THE INSTRUMENT TYPE
      IREC=IREC+1
      IF(ITYPE.EQ.1)CALL HEADR  ! IF "ERBE" RECORD, CALL HEADR
      IF(ITYPE.EQ.11)CALL PCM    ! IF "PCM" RECORD, CALL PCM
      IF(ITYPE.LT.5)GOTO1     ! OTHERWISE CHECK THE NEXT RECORD
      IF(ITYPE.EQ.11)GOTO1    ! IF IT WAS PCM GO BACK TO TOP FOR ANOTHER
      BACKSPACE 8
      READ(8,4,IOSTAT=IOS,ERR=98,END=99)(LINE(I),I=1,65)
4     FORMAT(2X,65A2)
      WRITE(6,9)LINE
9     FORMAT(1X,65A2)
      GOTO1
98    WRITE(6,7)IOS,ITYPE
7     FORMAT(" IOS=",I5," ITYPE= ",O4)
99    CONTINUE
      WRITE(LUT,1099)
1099  FORMAT("EOF ENCOUNTERED, ENTER OPTION:",/,
     1" 0= STOP",/," 1= CONTINUE BUT WITH NEW DESCRIPTION",/,
     1" 2= CONTINUE")
      READ(LUT,*)IOPT
      IF(IOPT.EQ.1)GOTO100
      IF(IOPT.EQ.2)GOTO1
      STOP
      END
```

```
      SUBROUTINE HEADR
C
C
C
C     THIS SUBROUTINE PROCESSES HEADR INFORMATION FOR "ERBE" (TIROS)
C         RECORDS.
C
C
      COMMON/BUF/INST,IANA
      DIMENSION KTIME(6),ITBUF(13),IDIST(2),IAZ(2),IEL(2)
      DIMENSION C14(6),C15(6)
      DIMENSION IBUF(3936),LBUF(3938)
      EQUIVALENCE(IDIST(1),DIST)
      EQUIVALENCE(IAZ(1),AZ)
      EQUIVALENCE(IEL(1),EL)
      DATA NREC/0/
      DATA C14/347.83511,-583.00496,489.0738,-211.33864,
     145.022096,-3.7944378/
      DATA C15/90.962177,-172.73195,168.50204,-90.571380,
     123.545091,-2.33822189/
      DATA MASK/777B/
      DATA MAS2/7777B/
      BACKSPACE 8
      NREC=NREC+1
      LUT=LOGLU(IDUM)
      LIM=1642
      IF(INST.EQ.1)LIM=3938
      LIM2=LIM-2
      CALL LGBUF(LBUF,LIM )
      READ(8,IOSTAT=IOS,ERR=98,END=99)ITYPE,INST,(IBUF(I),I=1,LIM2)
      MF=97
      IF(INST.EQ.0)MF=41
      NFRAM=IAND(IBUF(MF),MASK)
      IF(INST.EQ.0)GOTO48
      IDEL=39-NFRAM
      IF(IDEL.LT.0)IDEL=IDEL+320
      IPT=MF+IDEL*12+11
      IECH=IBUF(IPT)
      ISTAT=IBUF(IPT-1)
      GOTO49
48    IDEL=320-NFRAM
      IF(IDEL.GT.319)IDEL=0
      IPT=MF+IDEL*5+4
      IECH=IBUF(IPT)
      ISTAT=IBUF(IPT+5)
49    CONTINUE
      KTIME(6)=IBUF(IND(2,1))
      DO 50 K=1,5
      KTIME(K)=IBUF(IND(1,K))
50    CONTINUE
      CALL CNVTM(KTIME,ITBUF)
      IVA=IBUF(IND(6,3))
      IF(IVA.EQ.0)THEN
      VAC= 4HAIR
      ELSE
```

```
110         VAC =4HVAC
111         ENDIF
112         NVAL=IBUF(IND(7,2))
113         IF(NVAL.EQ.0)THEN
114         IVAL=2HN
115         ELSE
116         IVAL=2HY
117         ENDIF
118         IDIST(1)=IBUF(IND(2,3))
119         IDIST(2)=IBUF(IND(2,4))
120         IAZ(1)=IBUF(IND(6,4))
121         IAZ(2)=IBUF(IND(6,5))
122         IEL(1)=IBUF(IND(7,4))
123         IEL(2)=IBUF(IND(7,5))
124         INS=2HNS
125         INSTR=IBUF(IND(4,2))
126         IF(INST.EQ.1)INS=2HSC
127         IST=IBUF(IND(4,3))
128         MAJF=IBUF(IND(2,2))
129         WRITE(6,101)NREC,ITBUF,INS,IST,NVAL,IVA ,IECH,ISTAT,NFRAM,MAJF
130        1,INSTR
131  101    FORMAT(I5,2X,13A2,2X,A2,2X,A2,2X,I2,I3,
132        1" COMMAND ECHO ",I6,2X," STAT ",I6,2X," 1ST mf: ",I5,
133        1" MAJ F#: ",I5,A2)
134         RETURN
135  98     CONTINUE
136         WRITE(LUT,100)IOS
137  100    FORMAT(" ERROR ON TAPE=",I5)
138  99     RETURN
139         END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:  8358        COMMON: (NONE)
```

```
      SUBROUTINE PCM
C
C
C
C         THIS SUBROUTINE PROCESSES HEADR INFORMATION FOR "PCM" RECORDS
C
C
      DIMENSION IBUF(1700),JBUF( 7),LBUF(1700)
      EQUIVALENCE(IBUF(1),JBUF(1))
      BACKSPACE 8
      LUT=LOGLU(IDUM)
      CALL LGBUF(LBUF,1700)
      READ(8,IOSTAT=IOS,ERR=98,END=99)IBUF
      DO 80 L= 1,4
      K=(L-1)*400
      WRITE(6,100) IBUF(143+K),IBUF(293+K),IBUF(297+K),IBUF(306+K),
     *IBUF(310+K),IBUF(455+K),IBUF(468+K),JBUF(5),JBUF(4),JBUF(3),
     * JBUF(6),JBUF(7)
100   FORMAT(7I10,5X,5I6)
80    CONTINUE
      RETURN
98    WRITE(LUT,101)IOS
101   FORMAT(" ERROR # ",I5)
      RETURN
99    WRITE(LUT,102)
102   FORMAT(" EOF")
      STOP
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 3575 COMMON: (NONE)

```
 168          BLOCK DATA BUF
 169          COMMON/BUF/INST,IANA
 170          DATA INST/0/
 171          DATA IANA/0/
 172          END

FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM: (NONE)      COMMON: (NONE)

BLOCK COMMON BUF      SIZE:      2
```

```
  173       FUNCTION IND(I,J)
  174       COMMON /BUF/INST
  175       INT=5
  176       IF(INST.EQ.1)INT=12
  177       IND=(I-1)*INT+J
  178       RETURN
  179       END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 25 COMMON: (NONE)

TIP

TIP is a program used to convert TIP tapes (TIROS Information Processor) to TRW "ERBE" format tapes. Major frames of data are extracted from the TIP data stream for both the scanner and non-scanner instruments. The data is not merged, but output as separate major frames of non-scanner and scanner data with the convention that the scanner will always be designated "station A" and the non-scanner "station B". The tapes created by this TIP program can be processed using programs developed under this contract (such as MAGAN, SUMRY etc.) or processed by TRW software on the HP-1000 computer.

To run TIP, mount a TIP tape (generated at RCA during TIROS integration and testing) and then type TIP. Answer the computer generated questions appropriately.

```
$FILES(1,1)
      PROGRAM TIP
C     PROGRAM TIP IS DESIGNED TO READ TIP (TIROS INFORMATION
C     PROCESSOR) DATA TAPES AND EXTRACT ERBE INSTRUMENT DATA.
C     IT WILL REFORMAT THE ERBE DATA IN A FORM COMPATABLE WITH
C     THE TRW BCU SOFTWARE FOR DATA ANALYSIS.
C
C     MAJOR FRAMES OF DATA ARE EXTRACTED FROM THE TIP DATA FOR BOTH
C     THE SCANNER AND NON-SCANNER AT THE SAME TIME. THE DATA IS NOT
C     MIXED, BUT OUTPUT AS SEPARATE MAJOR FRAMES WITH THE CONVENTION
C     THAT THE SCANNER WILL ALWAYS BE DESIGNATED AS "STATION A" AND
C     THE NON-SCANNER AS "STATION B".
C
C
C
C
      DIMENSION IND(32),ICHAN(160),NCHAN(160),IPBUF(6)
      DIMENSION NAME(3),NAM(5),NAM2(5)    !NAME=TEST CONDUCTOR;NAM=OUTPUT FIL
      DIMENSION IBUF(1024),LBUF(1024),JBUF(53,10),LLBUF(3938)
      DIMENSION ISCAN(3938),NSCAN(1642),IHEAD(96),NHEAD(40)
      DIMENSION IFRAM(12,320),NFRAM(5,320)
      EQUIVALENCE(IHEAD(1),ISCAN(3)),(NHEAD(1),NSCAN(3))
C     ------------------------------------------------------------
C
C     ------------------------------------------------------------
      EQUIVALENCE(IFRAM(1),ISCAN(99)),(NFRAM(1),NSCAN(43))
      EQUIVALENCE(JBUF(1,1),IBUF(1))
      EQUIVALENCE (LBUF(1),LLBUF(1))
      DATA ISC,NSC/060000B,060000B/
      DATA IBOMB/1/
      DATA ISTR/1/
      DATA ISCF,NSCF/0,0/
      DATA LUIN,LUOUT/8,7/  !WHEN TAPE TO TAPE COPY, LUIN=9? AND LUOUT=8
      DATA IANS/1/
      DATA MASK/777B/                        !MINOR FRAME# MASK
      DATA IDIGB,NDIGB/7777B,7777B/          !DIGB START VALUES(SCANNER,NON-SCAN
      DATA M1,M2,M3,M4,M5,M6/177760B,17B,177400B,377B,170000B,7777B/
      DATA IB15,IB14,IB13,IB12/100000B,40000B,20000B,10000B/
      DATA IND/1,11,11,2,3,11,11,11,4,11,5,6,11,7,11,11,11,8,9,10,12*11/
      DATA ICHAN/4*1000B,0,7*1000B,400B,14*1000B,1400B,7*1000B,2000B,
     17*1000B,2400B,7*1000B,3000B,7*1000B,3400B,7*1000B,4000B,31*1000B,
     16000B,7*1000B,6400B,8*1000B,7000B,3*1000B,7400B,39*1000B/
      DATA NAM/'MTDFIL::21'/
      DATA NAM2/'MTDFIL::22'/
      DATA NCHAN/5*1000B,0,7*1000B, 400B,14*1000B,1400B,16*1000B,2400B,
     17*1000B,3000B,6*1000B,3400B,7*1000B,4000B,31*1000B,6000B,7*1000B,
     1 6400B,8*1000B,7000B,6*1000B,7400B,35*1000B/
      LUT=LOGLU(IDUM)           !GET LU OF OPERATOR'S TERMINAL
      CALL LGBUF(LBUF,1024)     !THE TIP RECORDS ARE 1024 WORDS LONG
      ISCAN(1)=1                !INDICATES INSTRUMENT DATA TO
C                               !TRW ERBE SOFTWARE
      NSCAN(1)=1                !SAME AS ABOVE LINE
      ISCAN(2)=1                    !INDICATES SCANNER
      NSCAN(2)=0                    !INDICATES NON-SCANNER
      IHEAD(38)=2HSC                !INDICATES SCANNER
```

```
      NHEAD(17)=2HNS                !INDICATES NON-SCANNER
      IHEAD(39)=2HA                 !SCANNER WILL ALWAYS BE STATION A
      NHEAD(18)=2HB                 !NON-SCANNER WILL ALWAYS BE STATION B
      IHEAD(14)=0                   !INITIALIZE MAJOR FRAME COUNTER(SCANNER)
      NHEAD(7)=0                    !INITIALIZE MAJOR FRAME COUNTER(NON-SCANNER
      IHEAD(63)=0                       !VAC FLAG (SET TO ZERO)
      IHEAD(74)=1                       !VALID DATA FLAG (SET TO 1 FOR VALID)
      NHEAD(23)=0                       !VAC FLAG FOR NON-SCANNER
      NHEAD(32)=1                       !VALID DATA FLAG NON-SCANNER
      WRITE(LUT,1000)
1000  FORMAT(" ENTER TEST PROCEDURE FOR SCANNER(12 CHARACTERS)")
      READ(LUT,1001)IHEAD(17),(IHEAD(KK),KK=25,29)
1001  FORMAT(6A2)
      WRITE(LUT,1005)
1005  FORMAT(" ENTER SERIAL NUMBER FOR SCANNER")
      READ(LUT,1007)IHEAD(61),IHEAD(62)
1007   FORMAT(2A2)

      WRITE(LUT,1002)
1002  FORMAT(" ENTER TEST PROCEDURE FOR NON-SCANNER(12 CHAR)")
      READ(LUT,1001)(NHEAD(KK),KK=10,15)
      WRITE(LUT,1008)
1008  FORMAT(" ENTER SERIAL NUMBER FOR NON SCANNER")
      READ(LUT,1007)NHEAD(26),NHEAD(27)
      WRITE(LUT,1003)
1003  FORMAT(" ENTER INITIALS OR NAME(6 CHARACTERS)")
      READ(LUT,1001)(NAME(KK),KK=1,3)
      IHEAD(49)=NAME(1)
      IHEAD(50)=NAME(2)
      IHEAD(51)=NAME(3)
      NHEAD(21)=NAME(1)
      NHEAD(22)=NAME(2)
      NHEAD(23)=NAME(3)
      WRITE(LUT,1004)
1004  FORMAT(" ENTER YEAR (4 DIGITS)")
      READ(LUT,*)NYR
      WRITE(LUT,1017)
1017  FORMAT(" DO YOU WANT TO COPY SCANNER DATA? (Y/N)")
      READ(LUT,1022)IJK
      IF(IJK.EQ.1HY)ISCF=1
      WRITE(LUT,1016)
1016  FORMAT(" DO YOU WANT TO COPY NON SCANNER DATA? (Y/N)" )
      READ(LUT,1022)IJK
      IF(IJK.EQ.1HY)NSCF=1
1019  WRITE(LUT,1021)
1021   FORMAT(" IS THIS A TAPE TO TAPE COPY (Y/N)")
      READ(LUT,1022)IJK
1022   FORMAT(A1)
      IF(IJK.EQ.1HN)GOTO1023
      IF(IJK.NE.1HY)GOTO1019
      LUIN=9
      OPEN(LUIN)
      LUOUT=8
      OPEN(LUOUT)
      WRITE(LUT,1024)
```

```
112 1024  FORMAT(" MOUNT SOURCE TAPE ON DYLON TAPE UNIT ",/,
113      1" MOUNT DESTINATION TAPE ON HP TAPE DRIVE" ,/,
114      1" TYPE GO WHEN READY ")
115       READ(LUT,1022)IJK
116       GOTO1
117 1023  CONTINUE
118       OPEN(LUIN)
119       OPEN(UNIT=LUOUT ,FILE=NAM,IOSTAT=IOS,ERR=1999)
120       WRITE(LUT,1997)
121 1997   FORMAT(" ÉHÉJ")
122       GOTO1
123 1999  WRITE(LUT,1998)IOS,NAM
124 1998  FORMAT(" ERROR#",I5," TRYING TO OPEN ",5A2)
125       STOP
126 1     NMF=0
127 10    READ(LUIN,ERR=98,END=67,IOSTAT=IOS)IBUF
128       GOTO68
129 67    WRITE(LUT,69)
130 69    FORMAT(" END OF INPUT TAPE, ENTER GO TO CONTINUE WITH ANOTHER",/
131      1" INPUT TAPE AND SAME OUTPUT TAPE (ANYTHING ELSE ABORTS)")
132       READ(LUT,1001)IOP
133       IF(IOP.NE.2HGO)GOTO887
134       GOTO10
135 68    CONTINUE
136       MF=IAND(MASK,IBUF(3))
137       IF(MF.EQ.NMF)GOTO12
138       WRITE(LUT,1011)NMF,MF
139 1011  FORMAT(" EXPECTING MINOR FRAME# ",I5," BUT FOUND MF#",I5)
140       WRITE(6,1011)NMF,MF
141       IF(MF.EQ.0)GOTO12
142       NMF=0                        !NEED TO SEARCH FOR MF=0 &START OVER
143       DO 11 N=2,10
144       MF=IAND(MASK,JBUF(3,N))
145       IF(MF.NE.0)GOTO11
146       MFF=1
147       MFL=1+10-N
148       K=N-1
149       GOTO13
150 11    CONTINUE
151       GOTO1                        !DIDN'T FIND MF=0 IN THIS GROUP, TRY AGAIN.
152 12    MFF=MF+1
153       MFL=MF+10
154       K=0
155       IF(MFL.GT.320)MFL=320
156 13    DO 990 M=MFF,MFL
157       K=K+1
158       IFRAM(1,M)=IOR(ISC,M-1)          !SCANNER MINOR FRAME#
159       IFRAM(5,M)=ISHFT(IAND(JBUF(10,K),M1),-4)
160       IFRAM(6,M)=IOR(ISHFT(IAND(JBUF(10,K),M2),+8),
161      1ISHFT(IAND(JBUF(15,K),M3),-8))
162       IFRAM(7,M)=IOR(ISHFT(IAND(JBUF(15,K),M4),+4),
163      1ISHFT(IAND(JBUF(23,K),M5),-12))
164       IFRAM(8,M)=IAND(JBUF(23,K),M6)
165       IFRAM(9,M)=ISHFT(IAND(JBUF(31,K),M1),-4)
166       IFRAM(10,M)=IOR(ISHFT(IAND(JBUF(31,K),M2),+8),
```

```
      1ISHFT(IAND(JBUF(37,K),M3),-8))
       IFRAM(11,M)=IOR(ISHFT(IAND(JBUF(37,K),M4),+4),
      1ISHFT(IAND(JBUF(44,K),M5),-12))
       IFRAM(12,M)=IAND(JBUF(44,K),M6)
       NFRAM(1,M)=IOR(NSC,M-1)      !NON-SCANNER MINOR FRAME#
       NFRAM(5,M)=JBUF(27,K)
C      DIGB SECTION FOLLOWS
       JB=JBUF(7,K)     !THE 7TH WORD OF EACH TIP MINOR FRAME FREQUENTLY
C                        CONTAINS ONE OR MORE "DIGITAL B" BIT VALUES
       GOTO(900,901,902,903,904,905,906,907,908,909,910)
      1IND(MOD(M-1,32)+1)     !THIS COMPUTED GOTO WILL SELECT THE APPROPRIATE
C                           ACTION ACCORDING TO THE MINOR FRAME NUMBER.
C                           THE "IND" ARRAY IS INITIALIZED WITH A DATA STATEMEN
C                           AND USED TO DIRECT THE TRAFFIC AS THE MINOR FRAMES
C                           ARE PROCESSED. THE PATTERN REPEATS EVERY 32 MINOR
C                           FRAMES, AND THEREFORE THE MODULUS EXPRESSION
C                           IN PARENS WILL EVALUATE TO ONE OF THE 32 VALUES
C                           FOR WHICH THERE IS A ENTRY IN "IND". ONLY 11
C                           BRANCHES ARE NEEDED TO HANDLE ALL CASES BECAUSE
C                           THERE ARE A NUMBER OF MINOR FRAMES WHICH DO NOT
C                           HAVE DIGITAL B DATA. SO "IND" HAS 32 ENTRIES IN
C                           IT, BUT ONLY 11 DIFFERENT VALUES.
C
900    IF(M.EQ.1)GOTO910          !ES10 SCAN MOTOR PWR (EXCEPT MINOR FRAME 0)
       NBIT=9
       IDIGB=IPOKE(IDIGB,NBIT,IAND(JB,IB12))
       GOTO910
901    NBIT=1                     !ES2 PULSE LOAD BUS-A
       IDIGB=IPOKE(IDIGB,NBIT,IAND(JB,IB14))
       NBIT=7                     !ES6 BLACKBODY HTR PWR
       IDIGB=IPOKE(IDIGB,NBIT,IAND(JB,IB13))
       NBIT=7                     !ENS6 BLACKBODY HTR PWR
       NDIGB=IPOKE(NDIGB,NBIT,IAND(JB,IB12))
       GOTO910
902    NBIT=8                     !ENS9 AZIMUTH MTR PWR
       NDIGB=IPOKE(NDIGB,NBIT,IAND(JB,IB15))

       NBIT=11                    !NS SPARE
       NDIGB=IPOKE(NDIGB,NBIT,IAND(JB,IB14))
       GOTO910
903    NBIT=4                     !SCANNER SPARE
       IDIGB=IPOKE(IDIGB,NBIT,IAND(JB,IB13))
       NBIT=10                    !PED STANDBY HTR PWR (SCANNER)
       IDIGB=IPOKE(IDIGB,NBIT,IAND(JB,IB12))
       GOTO910
904    NBIT=4                     !NS INST HTR PWR
       NDIGB=IPOKE(NDIGB,NBIT,IAND(JB,IB14))
       GOTO910
905    NBIT=9                     !NS ELEV. MTR PWR
       NDIGB=IPOKE(NDIGB,NBIT,IAND(JB,IB15))
       GOTO910
906    NBIT=2                     !SC PULSE LOAD BUS B
       IDIGB=IPOKE(IDIGB,NBIT,IAND(JB,IB14))
       GOTO910
907    NBIT=0                     !SC INSTR POWER
```

```
222       IDIGB=IPOKE(IDIGB,NBIT,IAND(JB,IB15))
223       NBIT=3                          !SC STANDBY HTR PWR
224       IDIGB=IPOKE(IDIGB,NBIT,IAND(JB,IB14))
225       NBIT=8                          !SC AZIMUTH MTR PWR
226       IDIGB=IPOKE(IDIGB,NBIT,IAND(JB,IB13))
227       NBIT=11                         !SC SPARE
228       IDIGB=IPOKE(IDIGB,NBIT,IAND(JB,IB12))
229       GOTO910
230 908   NBIT=0                          !NS INST PWR
231       NDIGB=IPOKE(NDIGB,NBIT,IAND(JB,IB15))
232       NBIT=1                          !NS PULSE LOAD BUS-A
233       NDIGB=IPOKE(NDIGB,NBIT,IAND(JB,IB14))
234       NBIT=2                          !NS PULSE LOAD BUS-B
235       NDIGB=IPOKE(NDIGB,NBIT,IAND(JB,IB13))
236       NBIT=3                          !NS HEAD STDBY HTR
237       NDIGB=IPOKE(NDIGB,NBIT,IAND(JB,IB12))
238       GOTO910
239 909   NBIT=10                         !NS PED STDBY HTR PWR
240       NDIGB=IPOKE(NDIGB,NBIT,IAND(JB,IB15))
241 910   IFRAM(2,M)=IDIGB    !UPDATE THE SCANNER DIG B WORD
242       NFRAM(2,M)=NDIGB        !UPDATE THE NON-SCANNER DIG B WORD
243       LOW=IAND(JB,M4)             !GET LOW BYTE OF WORD7
244 C                                     THIS WORD WILL SOMETIMES CONTAIN
245 C                                     ANALOG HOUSEKEEPING TELEMETRY DATA
246       JCHAN=MOD(M,160)
247       IF(JCHAN.EQ.0)JCHAN=1
248       IFRAM(3,M)=ICHAN(JCHAN)+LOW/4 + 200B
249       NFRAM(3,M)=NCHAN(JCHAN)+LOW/4 + 200B
250       IFRAM(4,M)=1000B            !BCU CHAN2 (DUMMY)
251       NFRAM(4,M)=1000B
252 990   CONTINUE
253       NMF=M-1
254       IF(NMF.LT.320)GOTO10
255       IHEAD(14)=IHEAD(14)+1
256       NHEAD(7)=NHEAD(7)+1
257       NDAY=IBUF(531)
258       NHR=IBUF(532)
259       NMIN=IBUF(533)
260       NSEC=IBUF(534)
261       NMSEC=0
262       IHEAD(1)=NMSEC
263       IHEAD(2)=NSEC
264       IHEAD(3)=NMIN
265       IHEAD(4)=NHR
266       IHEAD(5)=NDAY
267       IHEAD(13)=NYR
268       NHEAD(1)=NMSEC
269       NHEAD(2)=NSEC
270       NHEAD(3)=NMIN
271       NHEAD(4)=NHR
272       NHEAD(5)=NDAY
273       NHEAD(6)=NYR
274       CALL LGBUF(LLBUF,3938)
275       WRITE(LUT,1996)NHR,NMIN,NSEC,NMSEC,NDAY,NYR,IFRAM(11,40),
276      *IFRAM(12,40),NFRAM(5,2),NFRAM(5,1)
```

```
277         WRITE(6,1996)NHR,NMIN,NSEC,NMSEC,NDAY,NYR,IFRAM(11,40),
278        *IFRAM(12,40),NFRAM(5,2),NFRAM(5,1)
279 1996    FORMAT("   HR  MIN  SEC  NMSEC  DAY  YEAR SCAN STAT     ECHO",
280        *"  NSCAN STAT      ECHO",/,4I5,2I7,4I10)
281         IF(IFBRK(IDUM))1991,1990,1991
282 1991    WRITE(LUT,1992)
283 1992    FORMAT(" ENTER 1 TO START OR CONTINUE SAVING DATA"         ,/,
284        1" -1 TO STOP PROGRAM ",/,
285        1" 0 TO STOP SAVING DATA BUT CONTINUE READING ")
286         READ(LUT,*)IANS
287 1990    IF(IANS.EQ.-1)GOTO 887
288         IF(IANS.EQ.0)GOTO 886
289 2700    IF(ISCF.EQ.1)WRITE(UNIT=LUOUT,ERR=88,IOSTAT=IOS)ISCAN
290         IBOMB=2
291 2701    IF(NSCF.EQ.1)WRITE(UNIT=LUOUT,ERR=88,IOSTAT=IOS)NSCAN
292         IBOMB=1
293 886     NMF=0
294         IF(K.EQ.10)GOTO1
295         MFF=1
296         MFL=10-K
297         GOTO13       !FINISH WITH THE DATA IN THE BUFFER
298 88      ENDFILE(LUOUT)
299         CLOSE(LUOUT)
300         OPEN(UNIT=LUOUT,FILE=NAM2,IOSTAT=IOS,ERR=1999)
301         IF(IBOMB.EQ.1)GOTO2700
302         GOTO2701
303 887     ENDFILE LUOUT
304         ENDFILE LUOUT
305         CLOSE (LUOUT)
306         STOP
307 98      WRITE(LUT,988)IOS
308 988     FORMAT(" ERROR #",I5," ON TAPE FILE")
309 99      STOP
310         END


FTN4X COMPILER: HP92834 REV.2130 (810716)


**  NO WARNINGS **  NO ERRORS **   PROGRAM: 13042        COMMON: (NONE)
```

```
311         FUNCTION IPOKE(IWORD,NBIT,IBIT)
312 C
313 C       THIS FUNCTION WILL STASH A ZERO OR ONE (ACCORDING TO THE VALUE
314 C       OF IBIT) IN THE WORD IWORD AT BIT POSITION NBIT.
315 C       NBIT CAN BE ANY NUMBER FROM 0 TO 15. IBIT CAN BE 0 (ZERO) OR
316 C       NON ZERO.  A ZERO WILL CAUSE ONE BIT OF IWORD TO BE SET TO ZERO.
317 C       A NON ZERO VALUE WILL CAUSE ONE BIT OF IWORD TO BE SET TO ONE.
318         DATA MM,NN,JJ/177777B,077777B,100000B/
319         IF(NBIT.EQ.15)THEN
320         MAS=NN          !IF NBIT IS 15, SELECT MASK 'NN'
321         ELSE
322         MAS=MM-2**NBIT      !OTHERWISE, CONSTRUCT A MASK
323         ENDIF
324         IPOKE=IAND(IWORD,MAS)    !GRAB ALL THE BITS EXCEPT THE ONE MASKED OFF
325         IF(IBIT.EQ.0)RETURN      !IF IBIT IS ZERO, THE JOB IS DONE.
326         IF(NBIT.EQ.15)THEN
327         IPOKE=IOR(IPOKE,JJ)     !IF REPLACING 15TH BIT, USE MASK 'JJ'
328         ELSE
329         IPOKE=IOR(IPOKE,2**NBIT)  !OTHERWISE 'OR' IN THE BIT.
330         ENDIF
331         RETURN
332         END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 50 COMMON: (NONE)

PCM is a format conversion program. It is designed to read ERBS data tapes generated by Goddard during spacecraft passes and convert the data into an HP-1000 compatible format for data analysis.

```
$FILES(1,1)
      PROGRAM PCM
C
C
C
C                    PCM IS A FORMAT CONVERSION PROGRAM.  IT IS DESIGNED TO
C                    READ ERBS DATA TAPES GENERATED BY GODDARD DURING
C                    SPACE CRAFT PASSES AND CONVERT THE DATA INTO AN HP-1000
C                    COMPATABLE FORMAT FOR DATA ANALYSIS .
C
C
C
      DIMENSION IBUF(1700),NBUF(1508),LBUF(1700),IMF(10)
      INTEGER*4 ISECS
      EQUIVALENCE (ISECS,NBUF(87))
      COMMON/MSK/MSKR(16),MSKL(16)
      DATA LUIN,LUOUT/9,8/
      DATA IMF/2,11,21,30,39,51,60,70,80,89/
      DATA L12,L16/12,16/
      LUT=LOGLU(IDUM)
      CALL LGBUF(LBUF,1700)
      WRITE(LUT,2000)
2000  FORMAT(" MOUNT SOURCE ON DYLON, DEST ON HP DRIVE")
      IBUF(1)=11                    ! TRW RECORD TYPE 11 (PCM DATA)
      WRITE(LUT,1000)
1000  FORMAT(" ENTER TEST PROCEDURE (10 CHARS)" )
      READ(LUT,1001) (IBUF(J),J=9,13)
1001  FORMAT(5A2)
      WRITE(LUT,1002)
1002  FORMAT(" IS THIS TEST IN VACUUM? (Y/N)" )
      READ(LUT,1003)IANS
1003  FORMAT(A1)
      IBUF(14)= 0                   ! DEFAULT IS VACUUM
      IF(IANS.EQ.1HN) IBUF(14)=1    ! AIR
      WRITE(LUT,1004)
1004  FORMAT(" ENTER TEST CONDUCTOR'S INITIALS (6 CHARS)")
      READ(LUT,1001)(IBUF(J),J=15,17)
      WRITE(LUT,1005)
1005  FORMAT(" ENTER SCANNER SERIAL NUMBER (4 DIGITS)")
      READ(LUT,1001)IBUF(18),IBUF(19)
      WRITE(LUT,1006)
1006  FORMAT(" ENTER NON-SCANNER SERIAL NUMBER")
      READ(LUT,1001)IBUF(20),IBUF(21)
      WRITE(LUT,1007)
1007  FORMAT(" ENTER MAG TAPE REEL# (2 DIGITS)")
      READ(LUT,1008)IBUF(22)
1008  FORMAT(I2)
      WRITE(LUT,1030)
1030  FORMAT(" ENTER YEAR (4 DIGITS)")
      READ(LUT,1031)IBUF(7)
1031  FORMAT(I4)
      IBUF(23)=1                    ! INDICATES VALID DATA
      OPEN(LUIN,IOSTAT=IOS,ERR=1999)
      OPEN(LUOUT,IOSTAT=JOS,ERR=1998)
1     NMF=0                         ! NEXT EXPECTED MINOR FRAME #
```

```
 57 10    READ(LUIN,ERR=98,END=25,IOSTAT=IOS)NBUF
 58       GOTO30
 59 25    WRITE(LUT,1009)
 60 1009  FORMAT(" END OF INPUT TAPE, ENTER GO TO CONTINUE",/,
 61      1" WITH ANOTHER INPUT TAPE & SAME OUTPUT TAPE",/,
 62      1" ANY OTHER RESPONSE TERMINATES PROGRAM.")
 63       READ(LUT,1001)IANS
 64       IF(IANS.NE.2HGO)GOTO3999
 65       GOTO10
 66 30    MF=ISHFT(IAND(NBUF(2),MSKL(8)),-8)
 67       DO 100 I=1,16
 68       LPOS=8                          ! FIRST SCANNER DATA IS IN UPPER BYTE
 69       I100=I*100 + 1                  ! POINTS TO BEGINNING OF OUTPUT MINOR FRAME
 70       K=13 + (I-1)*94                 ! INDEX TO BEGINNING OF SCANNER DATA
 71       IBUF(I100)= NBUF(K-12)          ! MAJOR FRAME NUMBER
 72       IBUF(I100+1) = ISHFT(IAND(NBUF(K-11),MSKL(8)),-8)  ! MINOR FRAME #
 73       IBUF(I100+20) =       IAND(NBUF(K-1),MSKR(8))       ! DIG B
 74       IBUF(I100+69) = 0               ! NO ANALOG AVAILABLE FOR THIS SPOT
 75       IBUF(I100+98)= 175363B          ! SYNC WORD
 76       IBUF(I100+99)= 40B              ! SYNC WORD
 77       IBUF(I100+79) = IAND(ISHFT(IAND(NBUF(K-1),MSKR(8)),8),
 78      1 ISHFT(IAND(NBUF(K),MSKL(8)),-8))      ! ANALOG TELEMETRY
 79       IBUF(I100+48) = NBUF(K+73)             ! TIME CODE DATA
 80       IBUF(I100+49) = NBUF(K+74)             ! TIME CODE DATA
 81       IBUF(I100+50) = NBUF(K+75)             ! TIME CODE DATA
 82       DO 90 J=1,10                    ! INSTRUMENT MINOR FRAME LOOP
 83       LF=IMF(J)+I100                  ! START LOC IN IBUF FOR MINOR FRAME
 84       LL=LF+7                         ! STOP LOC IN IBUF FOR SCANNER DATA
 85 C     IBUF(LL+1)=IBITS(NBUF(K),LPOS,L16,K)  ! GET NON-SCANNER VALUE
 86       DO 80 L=LF,LL
 87 C     WRITE(6,2100)K,LPOS,NBUF(K),NBUF(K+1)
 88 2100   FORMAT(" K,LPOS,NBUF(K),NBUF(K+1): ",2I5,2O8)
 89       IBUF(L)= IBITS(NBUF(K),LPOS,L12,K)  ! GET SCANNER DATA VALUE
 90 C     WRITE(6,2101)L,IBUF(L),LPOS,K
 91 2101   FORMAT(" L,IBUF(L),LPOS,K:",I5,O8,2I5,//)
 92 80    CONTINUE
 93 C     WRITE(6,2102)K,LPOS,NBUF(K),NBUF(K+1)
 94 2102   FORMAT(" K,LPOS,NBUF(K),NBUF(K+1):"  ,2I5,2O8,////)
 95       IBUF(LL+1)= IBITS(NBUF(K),LPOS,L16,K)  ! GET NON-SCANNER VALUE
 96 90    CONTINUE
 97 100   CONTINUE
 98       IBUF(6)= IOR (ISHFT(IAND(NBUF(1),1B),13),
 99      1 ISHFT(IAND(NBUF(86),177770B),-3))          ! JULIAN DAY
100       NHREDS=IOR(ISHFT(IAND(NBUF(87),3B),8),ISHFT(IAND(NBUF(88),
101      1 MSKL(8)),-8))
102       WRITE(LUT,582)(NBUF(LG),LG=86,88)
103 582   FORMAT(3O8)
104       ISECS= ISHFT(IAND(ISECS,777776000B),-10)
105       WRITE(LUT,327)NBUF(87),NBUF(88)
106 327   FORMAT(" ISECS = ", 2O8 )
107       NHR=ISECS/3600     ! HOURS
108       ISECS=ISECS-NHR*3600
109       MINS=ISECS/60
110       NSECS=MOD(ISECS,60)
111       IBUF(2)=NHREDS
```

```
      IBUF(3)=NSECS
      IBUF(4)=MINS
      IBUF(5)=NHR
      WRITE(LUOUT,ERR=4999,IOSTAT=JOS)IBUF
      WRITE(LUT,2424)( IBUF(KJ),KJ=2,7)
2424  FORMAT(6I5)

      GOTO10
1999  WRITE(LUT,1010)
1010  FORMAT(" ERROR TRYING TO OPEN INPUT TAPE")
      STOP
1998  WRITE(LUT,1011)
1011  FORMAT(" ERROR TRYING TO OPEN OUTPUT TAPE")
      STOP
3999  CONTINUE
      ENDFILE(LUOUT)
C     ENDFILE(LUTOUT)
      STOP
4999  WRITE(LUT,5000)JOS
5000  FORMAT(" OUTPUT ERROR #:",I5)
      STOP
98    WRITE(LUT,1012)
1012  FORMAT(" ERROR READING INPUT TAPE")
      STOP
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 6187 COMMON: (NONE)

```
      FUNCTION IBITS(INP,LPOS,LEN,K)
C
C
C
C     FUNCTION IBITS IS USED TO EXTRACT VALUES EITHER WITHIN DATA WORDS
C     OR BETWEEN OVERLAPPING DATA WORDS.
C
C
C
      DIMENSION INP(2)
C     BITS ARE NUMBERED 1 TO 16 (RATHER THAN 0 TO 15)
      COMMON/MSK/MSKR(16),MSKL(16)
      IF(LPOS.GT.16.OR.LPOS.LT.1)STOP
      IF(LEN.GT.16.OR.LEN.LT.1)STOP
      IF(LPOS-LEN)10,20,30
10    ISHFL=LEN-LPOS
      ISHFR=16-LEN+LPOS
      IBITS=IOR(ISHFT(IAND(INP(1),MSKR(LPOS)),ISHFL)
     1,ISHFT(IAND(INP(2),MSKL(LEN-LPOS)),-ISHFR))
      K=K+1
      LPOS=LPOS-LEN+16
      RETURN
20    IBITS=IAND(INP(1),MSKR(LEN))
      K=K+1
      LPOS=16
      RETURN
30    ISHFR=LPOS-LEN
      IBITS=IAND(ISHFT(INP(1),-ISHFR),MSKR(LEN))
      LPOS=LPOS-LEN
      RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 129 COMMON: (NONE)

```
  168         BLOCK DATA
  169         COMMON/MSK/MSKR(16),MSKL(16)
  170         DATA MSKR/1,3,7,17B,37B,77B,177B,377B,777B,1777B,
  171        * 3777B,7777B,17777B,37777B,77777B,177777B/
  172         DATA MSKL/100000B,140000B,160000B,170000B,174000B,
  173        * 176000B,177000B,177400B,177600B,177700B,177740B,
  174        * 177760B,177770B,177774B,177776B,177777B/
  175         END

FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM: (NONE)      COMMON: (NONE)

BLOCK COMMON MSK      SIZE:     32
```

ERB

Program ERB is designed to convert ERB data in all experiment format to ERB data formatted like TRW BCU tapes with scanner and/or non scannner major frames. Scanner, if present, will always be on station A. Non scanner, if present, will always be on station B.

```
$FILES(2,2)
      PROGRAM ERB
C     PROGRAM ERBS IS DESIGNED TO CONVERT ERBS DATA IN ALL
C     EXPERIMENT FORMAT TO ERBS DATA FORMATTED LIKE TRW BCU
C     TAPES WITH SCANNER AND/OR NON SCANNER MAJOR FRAMES.
C     SCANNER, IF PRESENT, WILL ALWAYS BE ON STATION A.
C     NON SCANNER, IF PRESENT, WILL ALWAYS BE ON STATION B.
C
C
C
      DIMENSION NAME(3),NAM(5)              !NAME=TEST CONDUCTOR;NAM=OUTPUT FILE
      DIMENSION NPUT(2)
      DIMENSION IFRAM(12,320),NFRAM(5,320),IHEAD(96),NHEAD(40)
      DIMENSION IDCB(144),IDC(144)
      COMMON/STAT/LUT,LUIN,LUOUT,IBUF(100),MJF,MNF,NMJF,
     *NMNF,IPAK(22,256),ISCAN(3938),NSCAN(1642),ICHAN(32),
     *NCHAN(32),LBUF(3938)
      DIMENSION NUM(4)
      EQUIVALENCE(IHEAD(1),ISCAN(3)),(NHEAD(1),NSCAN(3))
      EQUIVALENCE(IDCB(1),ISCAN(1)),(IDC(1),ISCAN(145))
C     ------------------------------------------------------------
C
      EQUIVALENCE(IFRAM(1),ISCAN(99)),(NFRAM(1),NSCAN(43))
      COMMON /TIMS/NMSEC,NSEC,NMIN,NHR,NDAY,NYR
      DATA ISC,NSC/060000B,060000B/
      DATA IBOMB/1/
      DATA ISTR/1/
      DATA ISCF,NSCF/0,0/
C     DATA LUIN,LUOUT/8,8/  !WHEN TAPE TO TAPE COPY, LUIN=9  AND LUOUT=8
      DATA IANS/1/
      DATA IDIGB,NDIGB/7777B,7777B/        !DIGB START VALUES(SCANNER,NON-SCAN
      DATA NAM/'MTDFIL::20'/
      DATA NUM/'20212223'/
      LUT=LOGLU(IDUM)           !GET LU OF OPERATOR'S TERMINAL
      WRITE(LUT,1100)
1100  FORMAT(" DO YOU WANT TO COPY ALL EXP. DATA TO DISK? Y/N")
      READ(LUT,1022)IJK
      IF(IJK.EQ.1HN)GOTO1200
1103  WRITE(LUT,1101)
1101  FORMAT(" IF TAPE FILE IS LONG, YOU MAY WANT TO INSERT BLANK",
     1/," DISK CARTRIDGE....TYPE G WHEN READY OR A TO ABORT")
      READ(LUT,1022)IJK
      IF(IJK.EQ.1HA)STOP
      IF(IJK.NE.1HG)GOTO1103
      DO 1104 IJK=1,4
      CALL PURGE(IDCB,IERR,NAM)
1104  CONTINUE
      ISCAN(1)=1                !INDICATES INSTRUMENT DATA TO
      IFILE=1
C                               !TRW ERBE SOFTWARE
      NSCAN(1)=1                !SAME AS ABOVE LINE
      ISCAN(2)=1                    !INDICATES SCANNER
      NSCAN(2)=0                    !INDICATES NON-SCANNER
      IHEAD(38)=2HSC                !INDICATES SCANNER
      NHEAD(17)=2HNS                !INDICATES NON-SCANNER
```

```
      IHEAD(39)=2HA                !SCANNER WILL ALWAYS BE STATION A
      NHEAD(18)=2HB                !NON-SCANNER WILL ALWAYS BE STATION B
      IHEAD(14)=0                  !INITIALIZE MAJOR FRAME COUNTER(SCANNER)
      NHEAD(7)=0                   !INITIALIZE MAJOR FRAME COUNTER(NON-SCANNER)
      IHEAD(63)=0                      !VAC FLAG (SET TO ZERO)
      IHEAD(74)=1                      !VALID DATA FLAG (SET TO 1 FOR VALID)
      NHEAD(23)=0                      !VAC FLAG FOR NON-SCANNER
      NHEAD(32)=1                      !VALID DATA FLAG NON-SCANNER
      CALL INIT
      WRITE(LUT,1000)
1000  FORMAT(" ENTER TEST PROCEDURE FOR SCANNER(12 CHARACTERS)")
      READ(LUT,1001)IHEAD(17),(IHEAD(KK),KK=25,29)
1001  FORMAT(6A2)
      WRITE(LUT,1005)
1005  FORMAT(" ENTER SERIAL NUMBER FOR SCANNER")
      READ(LUT,1007)IHEAD(61),IHEAD(62)
1007   FORMAT(2A2)

      WRITE(LUT,1002)
1002  FORMAT(" ENTER TEST PROCEDURE FOR NON-SCANNER(12 CHAR)")
      READ(LUT,1001)(NHEAD(KK),KK=10,15)
      WRITE(LUT,1008)
1008  FORMAT(" ENTER SERIAL NUMBER FOR NON SCANNER")
      READ(LUT,1007)NHEAD(26),NHEAD(27)
      WRITE(LUT,1003)
1003  FORMAT(" ENTER INITIALS OR NAME(6 CHARACTERS)")
      READ(LUT,1001)(NAME(KK),KK=1,3)
      IHEAD(49)=NAME(1)
      IHEAD(50)=NAME(2)
      IHEAD(51)=NAME(3)
      NHEAD(21)=NAME(1)
      NHEAD(22)=NAME(2)
      NHEAD(23)=NAME(3)
      WRITE(LUT,1004)
1004  FORMAT(" ENTER YEAR (4 DIGITS)")
      READ(LUT,*)NYR
1019  WRITE(LUT,1021)
1021   FORMAT(" IS THIS A TAPE TO TAPE COPY (Y/N)")
      READ(LUT,1022)IJK
1022   FORMAT(A1)
      IF(IJK.EQ.1HN)GOTO1023
      IF(IJK.NE.1HY)GOTO1019
      LUIN=9
      OPEN(LUIN)
      LUOUT=8
      OPEN(LUOUT)
      WRITE(LUT,1024)
1024  FORMAT(" MOUNT SOURCE TAPE ON DYLON TAPE UNIT ",/,
     1" MOUNT DESTINATION TAPE ON HP TAPE DRIVE" ,/,
     1" TYPE GO WHEN READY ")
      READ(LUT,1022)IJK
      GOTO1
1023  CONTINUE
      LUIN=8
      LUOUT=10
```

```
      OPEN(LUIN)
      OPEN(UNIT=LUOUT ,FILE=NAM,IOSTAT=IOS,ERR=1999)
      WRITE(LUT,1997)
1997   FORMAT(" ESCH ESCJ")
      GOTO1
1999  WRITE(LUT,1998)IOS,NAM
1998  FORMAT(" ERROR#",I5," TRYING TO OPEN ",5A2)
      STOP
1     CONTINUE
      CALL INP(IOS)
      IF(IOS.NE.0)GOTO887
      IHEAD(1)=NMSEC
      IHEAD(2)=NSEC
      IHEAD(3)=NMIN
      IHEAD(4)=NHR
      IHEAD(5)=NDAY
      IHEAD(13)=NYR
      NHEAD(1)=NMSEC
      NHEAD(2)=NSEC
      NHEAD(3)=NMIN
      NHEAD(4)=NHR
      NHEAD(5)=NDAY
      NHEAD(6)=NYR
      CALL STUFF
      CALL LGBUF(LBUF ,3938)
      WRITE(LUT,1996)NHR,NMIN,NSEC,NMSEC,NDAY,NYR
1996  FORMAT("   HR  MIN  SEC  NMSEC  DAY  YEAR",/,6I5)
      IRET=0
87    WRITE(UNIT=LUOUT,ERR=88,IOSTAT=IOS)ISCAN
      IRET=1            ! IRET IS USED TO DETERMINE WHICH WRITE WAS
C                       ! EXECUTED LAST WHEN AN IOSTAT=533 OCCURRED.
86    WRITE(UNIT=LUOUT,ERR=88,IOSTAT=IOS)NSCAN
      GOTO1
88    ENDFILE(LUOUT)
      CLOSE(LUOUT)
      IF(IOS.NE.533)GOTO98
      IFILE=IFILE+1
      IF(IFILE.GT.4)GOTO887
      NAM(5)=NUM(IFILE)
      OPEN(UNIT=LUOUT,FILE=NAM,IOSTAT=IOS,ERR=1999)
      WRITE(LUT,1088)
1088  FORMAT(" EXTENDING MTDFIL TO NEXT AVAILABLE DISK LU")
      IF(IRET.EQ.0)GOTO87
      GOTO86
887   ENDFILE LUOUT
      ENDFILE LUOUT
      CLOSE (LUOUT)
      CLOSE(LUIN)
1200  WRITE(LUT,1201)
1201    FORMAT(" DO YOU WANT TO COPY DISK TO TAPE? Y/N")
      READ(LUT,1022)IJK
      IF(IJK.EQ.1HN)STOP
      IF(IJK.NE.1HY)GOTO1200
      WRITE(LUT,1202)
1202  FORMAT(" MOUNT TAPE WITH WRITE RING & TYPE G TO GO( A TO",
```

```
167        1" ABORT)")
168         READ(LUT,1022)IJK
169         IF(IJK.EQ.1HA)STOP
170         IF(IJK.NE.1HG)GOTO1200
171         CALL OPENF(IDC,IERR,8)
172         IF(IERR.GE.0)GOTO1491
173         WRITE(LUT,1490)IERR
174  1490   FORMAT(" ERROR# ",I5," TRYING TO OPEN TAPE UNIT FILE")
175         STOP
176  1491   DO 1492 IJK=1,IFILE
177         ICAR=19+IJK
178         CALL OPEN(IDCB,IERR,NAM,0,0,ICAR)
179         IF(IERR.LT.0)GOTO1499
180  1493    CALL READF(IDCB,IERR,IBUF,3938,ILEN)
181         IF(IBUF.EQ.0)GOTO1499
182         IF((IERR.LT.0).OR.(ILEN.LE.0))GOTO1494
183         CALL WRITF(IDC,IERR,IBUF,ILEN)
184         GOTO1493
185  1494   CALL CLOSE(IDCB)
186  1492   CONTINUE
187  1499    CALL EXEC(3,8+100B)
188         CALL EXEC(3,8+100B)
189         CALL RWNDF(IDC)
190         STOP
191  98     WRITE(LUT,988)IOS
192  988    FORMAT(" ERROR #",I5," ON DISK FILE")
193  99     STOP
194         END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:  1653        COMMON: (NONE)
```

```
      SUBROUTINE SC(IDAT)
      COMMON/STAT/LUT,LUIN,LUOUT,IBUF(100),MJF,MNF,NMJF,
     *NMNF,IPAK(22,256),ISCAN(3938),NSCAN(1642),ICHAN(32),
     *NCHAN(32),LBUF(3938)
      DIMENSION IFRAM(12,320),NFRAM(5,320)
      EQUIVALENCE(IFRAM(1),ISCAN(99)),(NFRAM(1),NSCAN(43))
      DATA J,K/1,5/
      IFRAM(K,J)=IDAT
      K=K+1
      IF(K.LT.13)RETURN
      K=5
      J=J+1
      IF(J.GT.320)J=1
      RETURN
      END
```

```
FTN4X COMPILER: HP92834 REV.2130 (810716)

**  NO WARNINGS **  NO ERRORS **   PROGRAM:     46        COMMON: (NONE)
```

```
      SUBROUTINE NS(IDAT)
      COMMON/STAT/LUT,LUIN,LUOUT,IBUF(100),MJF,MNF,NMJF,
     *NMNF,IPAK(22,256),ISCAN(3938),NSCAN(1642),ICHAN(32),
     *NCHAN(32),LBUF(3938)
      DIMENSION NFRAM(5,320)
      EQUIVALENCE(NFRAM(1),NSCAN(43))
      DATA J/1/
      NFRAM(5,J)=IDAT
      J=J+1
      IF(J.GT.320)J=1
      RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 28 COMMON: (NONE)

```
      SUBROUTINE INIT
      COMMON/STAT/LUT,LUIN,LUOUT,IBUF(100),MJF,MNF,NMJF,
     *NMNF,IPAK(22,256),ISCAN(3938),NSCAN(1642),ICHAN(32),
     *NCHAN(32),LBUF(3938)
      DIMENSION IFRAM(12,320),NFRAM(5,320)
      EQUIVALENCE(IFRAM(1),ISCAN(99)),(NFRAM(1),NSCAN(43))
      DATA ICHAN2/1000B/
      DO 100 I=1,320
      MNF=60000B+I-1
      IFRAM(1,I)=MNF
      NFRAM(1,I)=MNF
      IFRAM(3,I)=ICHAN2
      IFRAM(4,I)=ICHAN2
      NFRAM(3,I)=ICHAN2
      NFRAM(4,I)=ICHAN2
  100 CONTINUE
      RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 66 COMMON: (NONE)

```
241         SUBROUTINE INP(IOS)
242         COMMON/STAT/LUT,LUIN,LUOUT,IBUF(100),MJF,MNF,NMJF,
243        *NMNF,IPAK(22,256),ISCAN(3938),NSCAN(1642),ICHAN(32),
244        *NCHAN(32),LBUF(3938)
245         DATA IUPR/177400B/
246         DATA IDBUG/1/
247 C       IF(IDBUG.EQ.0)GOTO1
248 C       WRITE(LUT,102)
249 C02     FORMAT(" ENTER 1 TO START/CONTINUE DBUG PRINTOUT")
250 C       READ(LUT,*)IANS
251 C       IF(IANS.NE.1)THEN
252 C       IDBUG=0
253 C       GOTO1
254 C       ENDIF
255 C       WRITE(6,104)
256 C04     FORMAT(///)
257 1       CALL START(IOS)
258         IF(IOS.NE.0)RETURN
259         CALL TIM(IBUF)
260         I=0
261 2       I=I+1
262         IPAK(1,I)=IBUF(1)            ! COMMON AREA #1
263         IPAK(2,I)=IBUF(2)             !       "
264         IPAK(3,I)=IBUF(3)             !       "
265         IPAK(4,I)=IBITS(IBUF(13),8,16)   ! NON SCANNER DIG A
266         IPAK(5,I)=IBITS(IBUF(14),8,12)   ! SCANNER DIG A
267         IPAK(6,I)=IBITS(IBUF(15),12,12)  ! SCANNER DIG A
268         IPAK(7,I)=IBITS(IBUF(34),8,12)     ! SCANNER DIG A
269         IPAK( 8,I)=IBITS(IBUF(35),12,12)   ! SCANNER DIG A
270         IPAK( 9,I)=IBITS(IBUF(40),8,8)     ! SCANNER & NON SC. DIG B
271         IPAK(10,I)=IBUF(51)      ! COMMON #2
272         IPAK(11,I)=IBUF(52)      ! COMMON #2
273         IPAK(12,I)=IBUF(53)      ! COMMON #2
274         IPAK(13,I)=IBITS(IBUF(59),8,12)   ! SCANNER DIG A
275         IPAK(14,I)=IBITS(IBUF(60),12,12)   ! SCANNER DIG A
276         IPAK(15,I)=IBUF(62)  ! NON SC. DIG A ( STARTS MNF# 3, THEN EVERY 4TH)
277         IPAK(16,I)=IBITS(IBUF(74),8,12)  ! SCANNER DIG A
278         IPAK(17,I)=IBITS(IBUF(75),12,12)    ! SCANNER DIG A
279         IPAK(18,I)=IBITS(IBUF(77),16,8)   ! SCANNER ANALOG TELE.
280         IPAK(19,I)=IBITS(IBUF(77),8,8)     ! SC & NON-SC ANALOG TEL.
281         IPAK(20,I)=IBITS(IBUF(78),16,8)  ! SC & NON SC DIG B
282         IPAK(21,I)=IBITS(IBUF(98),16,12)  ! SCANNER DIG A
283         IPAK(22,I)=IBITS(IBUF(98),4,12)   ! SCANNER DIG A
284 C       IF(IDBUG.EQ.1)WRITE(6,103)(J,IPAK(J,I),J=1,27)
285 C03     FORMAT(I5,07)
286         IF(I.EQ.256)RETURN
287 C       READ(LUIN,ERR=98,END=99,IOSTAT=IOS)IBUF
288         CALL READIT(IOS)
289         IF(IOS.NE.0)RETURN
290         MJ=ISHFT(IAND(IBUF(1),IUPR),-8)
291         IF(MJ.NE.NMJF)THEN
292         WRITE(LUT,100)NMJF,MJ
293 100     FORMAT(" EXPECTING MAJOR FRAME#",I5," BUT FOUND #",I5)
294         GOTO1
295         ENDIF
```

```
296         MNF=ISHFT(IAND(IBUF(2),IUPR),-8)
297         IF(NMNF.NE.MNF)THEN
298         WRITE(LUT,101)NMNF,MNF
299  101   FORMAT(" EXPECTING MINOR FRAME# ",I5," BUT FOUND # ",I5)
300         GOTO1
301         ENDIF
302         NMNF=NMNF+1
303         IF(NMNF.GT.31)THEN
304         NMNF=0
305         MJF=MJ+1
306         NMJF=NMJF+1
307         IF(NMJF.GT.255)NMJF=0
308         IF(MJF.GT.255)MJF=0
309         ENDIF
310         GOTO2
311         END


FTN4X COMPILER: HP92834 REV.2130 (810716)

**INP   **WARNING 68 DETECTED AT LINE  311

READIT SHORTENED TO READT

**    1 WARNING  **  NO ERRORS **   PROGRAM:   410       COMMON: (NONE)
```

```
312       SUBROUTINE TIM(IBUF)
313       DIMENSION IBUF(100)
314       COMMON /TIMS/ NMSEC,NSEC,NMIN,NHR,NDAY,NYR
315       NDAY=ISHFT(IAND(IBUF(1),1),13)  ! GET MSB OF DAY
316       NDAY= IOR(NDAY,ISHFT(IAND(IBUF(51),177770B),-3))
317       NDAY=NDAY-5699
318       SEC=FLOAT(IAND(IBUF(51),7B))
319       SEC=SEC*16384+FLOAT(ISHFT(IAND(IBUF(52),177774B),-2))
320       MILSEC=IOR(ISHFT(IAND(IBUF(52),3B),8),ISHFT(IAND(IBUF(53),
321      *177400B),-8))
322       NMSEC=MILSEC/100
323       NHR=INT(SEC/3600.)
324       NMIN=INT(SEC/60.)-NHR*60
325       NSEC=INT(AMOD(SEC,60.))
326       RETURN
327       END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:    118       COMMON: (NONE)
```

```
328         SUBROUTINE START(IOS)
329         COMMON/STAT/LUT,LUIN,LUOUT,IBUF(100),MJF,MNF,NMJF,
330        *NMNF,IPAK(22,256),ISCAN(3938),NSCAN(1642),ICHAN(32),
331        *NCHAN(32),LBUF(3938)
332         DATA IUPR,ILWR/177400B,377B/
333         DATA MAJ /3B/
334         NMNF=1
335 C1      READ(LUIN,ERR=98,END=99,IOSTAT=IOS)IBUF
336 1       CALL READIT(IOS)
337         IF(IOS.NE.0)RETURN
338         MJF=ISHFT(IAND(IBUF(1),IUPR),-8)
339         IFLG=IAND(MJF,MAJ)
340         IF(IFLG.NE.0)GOTO1
341         NMJF=MJF
342 2       MNF=ISHFT(IAND(IBUF(2),IUPR),-8)
343         IF(MNF.EQ.0)RETURN
344 C       READ(LUIN,ERR=98,END=99,IOSTAT=IOS)IBUF
345         CALL READIT(IOS)
346         IF(IOS.NE.0)RETURN
347         IFLG=ISHFT(IAND(IBUF(1),IUPR),-8)
348         IF(IFLG.NE.MJF)GOTO1
349         GOTO2
350         END


FTN4X COMPILER: HP92834 REV.2130 (810716)


 **START **WARNING 68 DETECTED AT LINE  350

 READIT SHORTENED TO READT

 **    1 WARNING   **  NO ERRORS **   PROGRAM:     59        COMMON: (NONE)
```

```
351         SUBROUTINE READIT(IOS)
352         COMMON/STAT/LUT,LUIN,LUOUT,IBUF(100),MJF,MNF,NMJF,
353        *NMNF,IPAK(22,256),ISCAN(3938),NSCAN(1642),ICHAN(32),
354        *NCHAN(32),LBUF(3938)
355         DIMENSION KBUF(800),MBUF(800)
356         EQUIVALENCE(KBUF(1),LBUF(1)),(MBUF(1),LBUF(801))
357         DATA M/1/
358         CALL LGBUF(MBUF,800)
359         IF(M.EQ.1)READ(LUIN,ERR=98,END=99,IOSTAT=IOS)KBUF
360         ISTART=(M-1)*100+1
361         ISTOP=ISTART+99
362         J=0
363         DO 10 I=ISTART,ISTOP
364         J=J+1
365         IBUF(J)=LBUF(I)
366 10      CONTINUE
367         M=M+1
368         IF(M.GT.8)M=1
369         RETURN
370 98      WRITE(LUT,101)IOS
371 101     FORMAT(" ERROR# ",I5," READING TAPE")
372         RETURN
373 99      WRITE(LUT,102)
374 102     FORMAT(" EOF ON TAPE")
375         RETURN
376         END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **READIT**WARNING 68 DETECTED AT LINE  376

 READIT SHORTENED TO READT

 **    1 WARNING  **  NO ERRORS **   PROGRAM:    132       COMMON: (NONE)
```

```
377         BLOCK DATA
378         COMMON/STAT/LUT,LUIN,LUOUT,IBUF(100),MJF,MNF,NMJF,
379        *NMNF,IPAK(22,256),ISCAN(3938),NSCAN(1642),ICHAN(32),
380        *NCHAN(32),LBUF(3938)
381         COMMON/TIMS/NMSEC,NSEC,NMIN,NHR,NDAY,NYR
382         DATA ICHAN/0,7000B,13*1000B,2000B,7400B,1400B,1000B,400B,
383        *1000B,2400B,1000B,4000B,3400B,3000B,3*1000B,6000B,1000B,
384        *6400B/
385         DATA NCHAN/8*1000B,7400B,1400B,1000B,7000B,1000B,2400B,1000B,
386        *4000B,3400B,3000B,1000B,0,400B,6000B,1000B,6400B,8*1000B/
387         DATA LUIN,LUOUT/8,9/
388         END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM: (NONE)      COMMON: (NONE)

BLOCK COMMON TIMS    SIZE:     6

BLOCK COMMON STAT    SIZE: 15321
```

```
      SUBROUTINE STUFF
      COMMON/STAT/LUT,LUIN,LUOUT,IBUF(100),MJF,MNF,NMJF,
     *NMNF,IPAK(22,256),ISCAN(3938),NSCAN(1642),ICHAN(32),
     *NCHAN(32),LBUF(3938)
      DIMENSION IFRAM(12,320),NFRAM(5,320)
      EQUIVALENCE(IFRAM(1),ISCAN(99)),(NFRAM(1),NSCAN(43))
      DATA IDIGB,NDIGB/2*7777B/
      DO 100 I=1,256
      CALL NS(IPAK(4,I))
      CALL SC(IPAK(5,I))
      CALL SC(IPAK(6,I))
      CALL SC(IPAK(7,I))
      CALL SC(IPAK( 8,I))
      CALL SC(IPAK(13,I))
      CALL SC(IPAK(14,I))
      CALL SC(IPAK(16,I))
      CALL SC(IPAK(17,I))
      CALL SC(IPAK(21,I))
      CALL SC(IPAK(22,I))
      IF(MOD(I,4).EQ.0)CALL NS(IPAK(15,I))
      MN=MOD(I-1,32)+1
      J=I/.8
      K=18
      IF(MN.LT.3)K=19
      IF(ICHAN(MN).NE.1000B)IFRAM(3,J)=IPAK(K,I)+ICHAN(MN)
      IF(NCHAN(MN).NE.1000B)NFRAM(3,J)=IPAK(19,I)+NCHAN(MN)
100   CONTINUE
C
C
C
      DO 200 I=1,320
      J=I*.8
      J=MOD(J,32)
      IF(J.EQ.26)CALL DIGB(IDIGB,NDIGB,IPAK( 9,25),IPAK( 9,26),
     *IPAK( 9,27))
      IFRAM(2,I)=IDIGB
      NFRAM(2,I)=NDIGB
200   CONTINUE
      RETURN
      END
```

```
FTN4X COMPILER: HP92834 REV.2130 (810716)

**  NO WARNINGS **  NO ERRORS **   PROGRAM:   264       COMMON: (NONE)
```

```
      SUBROUTINE DIGB(IDIGB,NDIGB,I34,I35,I36)
      DATA L34,L35,L36/3*177777B/
      IF(I34.NE.L34) CALL S34(IDIGB,I34)
      IF(I35.NE.L35) CALL S35(IDIGB,NDIGB,I35)
      IF(I36.NE.L36) CALL S36(NDIGB,I36)
      L34=I34
      L35=I35
      L36=I36
      RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 41 COMMON: (NONE)

```
 439        BLOCK DATA
 440        COMMON/BITS/IBIT(8),NPOS(8),IPOS(8),JPOS(4)
 441        DATA IBIT/1,2,4,8,16,32,64,128/
 442        DATA IPOS/7,4,3,2,1,5,10,0/
 443        DATA NPOS/7,4,3,2,1,5,10,0/
 444        DATA JPOS/8,9, 8,9 /
 445        END
```

```
FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM: (NONE)      COMMON: (NONE)

BLOCK COMMON BITS    SIZE:    28
```

```
      SUBROUTINE S34(IDIGB,I34)
      COMMON/BITS/IBIT(8),NPOS(8),IPOS(8),JPOS(4)
      DO 10 I=1,8
      IB=IAND(IBIT(I),I34)
      J=IPOS(I)
      IDIGB=IPOKE(IDIGB,J,IB)
10    CONTINUE
      RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 36 COMMON: (NONE)

```
      SUBROUTINE S35(IDIGB,NDIGB,I35)
      COMMON/BITS/IBIT(8),NPOS(8),IPOS(8),JPOS(4)
      IB=IAND(IBIT(7),I35)
      J=JPOS(1)
      IDIGB=IPOKE(IDIGB,J,IB)
      IB=IAND(IBIT(8),I35)
      J=JPOS(2)
      IDIGB=IPOKE(IDIGB,J,IB)
      IB=IAND(IBIT(5),I35)
      J=JPOS(3)
      NDIGB=IPOKE(NDIGB,J,IB)
      IB=IAND(IBIT(6),I35)
      J=JPOS(4)
      NDIGB=IPOKE(NDIGB,J,IB)
      RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 54 COMMON: (NONE)

```
      SUBROUTINE S36(NDIGB,I36)
      COMMON/BITS/IBIT(8),NPOS(8),IPOS(8),JPOS(4)
      DO 10 I=1,8
      IB=IAND(IBIT(I),I36)
      J=NPOS(I)
      NDIGB=IPOKE(NDIGB,J,IB)
10    CONTINUE
      RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 36 COMMON: (NONE)

```
      FUNCTION IBITS(INP,LPOS,LEN)
      DIMENSION INP(2)
      COMMON/MSK/MSKR(16),MSKL(16)
      IF(LPOS.GT.16.OR.LPOS.LT.1)STOP
      IF(LEN.GT.16.OR.LEN.LT.1)STOP
      IF(LPOS-LEN)10,20,30
10    ISHFL=LEN-LPOS
      ISHFR=16-LEN+LPOS
      IBITS=ISHFT(IAND(INP(1),MSKR(LPOS)),ISHFL)
     1.OR.ISHFT(IAND(INP(2),MSKL(LEN-LPOS)),-ISHFR)
      RETURN
20    IBITS=IAND(INP(1),MSKR(LEN))
      RETURN
30    ISHFR=LPOS-LEN
      IBITS=IAND(ISHFT(INP(1),-ISHFR),MSKR(LEN))
      RETURN
      END
```

FTN4X COMPILER: HP92834 REV.2130 (810716)

** NO WARNINGS ** NO ERRORS ** PROGRAM: 111 COMMON: (NONE)

```
 497        BLOCK DATA
 498        COMMON/MSK/MSKR(16),MSKL(16)
 499        DATA MSKR/1,3,7,17B,37B,77B,177B,377B,777B,1777B,
 500       * 3777B,7777B,17777B,37777B,77777B,177777B/
 501        DATA MSKL/100000B,140000B,160000B,170000B,174000B,
 502       * 176000B,177000B,177400B,177600B,177700B,177740B,
 503       * 177760B,177770B,177774B,177776B,177777B/
 504        END

FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM: (NONE)      COMMON: (NONE)

BLOCK COMMON MSK      SIZE:    32
```

```
      FUNCTION IPOKE(IWORD,NBIT,IBIT)
C
C     THIS FUNCTION WILL RETURN A ZERO OR ONE (ACCORDING TO THE VALUE
C     OF IBIT) IN THE WORD IPOKE AT BIT POSITION NBIT.
C     NBIT CAN BE ANY NUMBER FROM 0 TO 15. IBIT CAN BE 0 (ZERO) OR
C     NON ZERO.  A ZERO WILL CAUSE ONE BIT OF IWORD TO BE SET TO ZERO.
C     A NON ZERO VALUE WILL CAUSE ONE BIT OF IWORD TO BE SET TO ONE.
      DATA MM,NN,JJ/177777B,077777B,100000B/
      IF(NBIT.EQ.15)THEN
      MAS=NN        !IF NBIT IS 15, SELECT MASK 'NN'
      ELSE
      MAS=MM-2**NBIT     !OTHERWISE, CONSTRUCT A MASK
      ENDIF
      IPOKE=IAND(IWORD,MAS)    !GRAB ALL THE BITS EXCEPT THE ONE MASKED OFF
      IF(IBIT.EQ.0)RETURN      !IF IBIT IS ZERO, THE JOB IS DONE.
      IF(NBIT.EQ.15)THEN
      IPOKE=IOR(IPOKE,JJ)     !IF REPLACING 15TH BIT, USE MASK 'JJ'
      ELSE
      IPOKE=IOR(IPOKE,2**NBIT)  !OTHERWISE 'OR' IN THE BIT.
      ENDIF
      RETURN
      END


FTN4X COMPILER: HP92834 REV.2130 (810716)

**  NO WARNINGS **  NO ERRORS **   PROGRAM:     50        COMMON: (NONE)
```

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-172567		

4. Title and Subtitle	5. Report Date
EARTH RADIATION BUDGET EXPERIMENT SOFTWARE DEVELOPMENT	April 1985
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
W. L. Edmonds	
	10. Work Unit No.
9. Performing Organization Name and Address	11. Contract or Grant No.
Systems and Applied Sciences Corporation 17 Research Drive Hampton, VA 23666	NAS1-16571
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code 619-12-01-01

15. Supplementary Notes

Langley Technical Monitor: Dwayne Hinton
Final Report

16. Abstract

Computer programming and analysis efforts during this three phase contract were carried out in support of the Earth Radiation Budget Experiment (ERBE) at NASA/Langley. The Earth Radiation Budget Experiment is described as well as data acquisition, analysis and modelling support for the testing of ERBE instruments. Also included are descriptions of the programs developed to analyze, format and display data collected during testing of the various ERBE instruments. Listings of the major programs developed under this contract are located in an appendix.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
ERBE, ERBS, radiation, weather, modelling	Unclassified-Unlimited Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	87	

For sale by the National Technical Information Service, Springfield, Virginia 22161